FORM PTO-1390 (REV. 1-98) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY'S DOCKET NUMBER 3726 US
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371	U.S. APPLICATION NO. (If known, see 37 CFR 1.5) 09/284113 Not Yet Known

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/US99/01988	28 January 1999	30 January 1998

TITLE OF INVENTION
System And Method For Creating And Manipulating Information Containers With Dynamic Registers

APPLICANT(S) FOR DO/EO/US
Michael De Angelo

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☑ This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.
2. □ This is a SECOND or SUBSEQUENT submission of items concerning a filing under 35 U.S.C. 371.
3. ☑ This express request to begin national examination procedures (35 U.S.C. 371 (f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. □ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☑ A copy of the International Application as filed (35 U.S.C. 371(c)(2)).
   - a. □ is transmitted herewith (required only if not transmitted by the International Bureau).
   - b. □ has been transmitted by the International Bureau.
   - c. ☑ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. □ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☑ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3)).
   - a. □ are transmitted herewith (required only if not transmitted by the International Bureau).
   - b. □ have been transmitted by the International Bureau.
   - c. ☑ have not been made; however, the time limit for making such amendments has NOT expired.
   - d. □ have not been made and will not be made.
8. □ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☑ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. □ A translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern document(s) or information included:

11. □ An Information Disclosure Statement under 37 CFR 1.97 and 1.9
12. ☑ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. □ A FIRST preliminary amendment.
    □ A SECOND or SUBSEQUENT preliminary amendment.
14. □ A substitute specification.
15. □ A change of power of attorney and/or address letter.
16. ☑ Other items or information: A Verified Statement Claiming Small Entity Status

U.S. APPLICATION NO. (if known. see 37 CFR 1.5)	INTERNATIONAL APPLICATION NO.	ATTORNEY'S DOCKET NUMBER
Not Yet Known	PCT/US99/01988	3726 US

17. ☑ The following fees are submitted:

CALCULATIONS PTO USE ONLY

BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(5)):

Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO.........$970.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO..$840.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO........$760.00

International preliminary examination fee (37 CFR 1.482) paid to USPTO but all claims did not satisfy provisions of PCT Article 33(2)-(4)......$670.00

International preliminary examination fee (37 CFR 1.482) paid to USPTO and all claims satisfied provisions of PCT Article 33(2)-(4).............$96.00

ENTER APPROPRIATE BASIC FEE AMOUNT =				$760.00	
Surcharge of $130.00 for furnishing the oath or declaration later than □ 20 □ 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				$0	
CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total claims	36 - 20 =	16	x $18.00	$288.00	
Independent claims	3 - 3 =	0	x $78.00	$0	
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+ $260.00	$0	
TOTAL OF ABOVE CALCULATIONS =				$1048.00	
Reduction of 1/2 for filing by small entity, if applicable. A Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28). +				$524.00	
SUBTOTAL =				$524.00	
Processing fee of $130.00 for furnishing the English translation later than □ 20 □ 30 months from the earliest claimed priority date (37 CFR 1.492(f)).				$0	
TOTAL NATIONAL FEE =				$524.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). $40.00 per property +				$40.00	
TOTAL FEES ENCLOSED =				$564.00	
				Amount to be rendered:	$564.00
				charged:	$

a. ☑ A check in the amount of $ 564.00 to cover the above fees is enclosed.

b. □ Please charge my Deposit Account No. ______________ in the amount of $ ____________ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☑ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 19-2555. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137 (a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Greg T. Sueoka
FENWICK & WEST LLP

Two Palo Alto Square
Palo Alto, CA 94306

SIGNATURE

Greg T. Sueoka
NAME

33,800
REGISTRATION NUMBER

COPY

U.S. APPLICATION NO. (if known. see 37 CFR 1.5) Not Yet Known	INTERNATIONAL APPLICATION NO. PCT/US99/01988	ATTORNEY'S DOCKET NUMBER 3726 US

17. ☑ The following fees are submitted: BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(5)): Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO.........$970.00 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO..$840.00 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO........$760.00 International preliminary examination fee (37 CFR 1.482) paid to USPTO but all claims did not satisfy provisions of PCT Article 33(2)-(4)......$670.00 International preliminary examination fee (37 CFR 1.482) paid to USPTO and all claims satisfied provisions of PCT Article 33(2)-(4).............$96.00				CALCULATIONS PTO USE ONLY	
ENTER APPROPRIATE BASIC FEE AMOUNT =				$760.00	
Surcharge of $130.00 for furnishing the oath or declaration later than □ 20 □ 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				$0	
CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total claims	36 - 20 =	16	x $18.00	$288.00	
Independent claims	3 - 3 =	0	x $78.00	$0	
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+ $260.00	$0	
TOTAL OF ABOVE CALCULATIONS =				$1048.00	
Reduction of 1/2 for filing by small entity, if applicable. A Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28). +				$524.00	
SUBTOTAL =				$524.00	
Processing fee of $130.00 for furnishing the English translation later than □ 20 □ 30 months from the earliest claimed priority date (37 CFR 1.492(f)).				$0	
TOTAL NATIONAL FEE =				$524.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). $40.00 per property +				$40.00	
TOTAL FEES ENCLOSED =				$564.00	
				Amount to be rendered:	$564.00
				charged:	$

a. ☑ A check in the amount of $ 564.00 to cover the above fees is enclosed.

b. □ Please charge my Deposit Account No. ______________ in the amount of $ ____________ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☑ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 19-2555. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137 (a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Greg T. Sueoka
FENWICK & WEST LLP

Two Palo Alto Square
Palo Alto, CA 94306

SIGNATURE

Greg T. Sueoka
NAME

33,800
REGISTRATION NUMBER

PTO/SB/ 10 (6-95) (modified)
Approved for use through 07/31/96. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(c))--SMALL BUSINESS CONCERN	Docket Number (Optional): 3726

Applicant or Patentee: Michael De Angelo

Application or Patent No.:

Filing Date or Issue Date:

Title: System And Method For Creating And Manipulating Information Containers With Dynamic Registers

I hereby declare that I am

[ ] the owner of the small business concern identified below:

[X] an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN Ematrix Corporation

ADDRESS OF SMALL BUSINESS CONCERN 104 West Anapamu, Suite C
Santa Barbara, California 93101

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

[X] the specification filed herewith with title as listed above.

[ ] the application identified above.

[ ] the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each such person, concern or organization having any rights in the invention is listed below:

[X] No such person, concern, or organization exists.

[ ] Each such person, concern or organization is listed below:

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING Michael De Angelo

TITLE OF PERSON IF OTHER THAN OWNER Officer

ADDRESS OF PERSON SIGNING 104 West Anapamu, Suite C, Santa Barbara, California 93101

SIGNATURE Michael DeAngelo DATE April 5, 1999

Rev. 04/01/99

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to computer systems in a multi-user mainframe or mini computer system, a client server network, or in local, wide area or public networks, and in particular, to computer networks for creating and manipulating information containers with dynamic interactive registers in a computer, media or publishing network, in order to manufacture information on, upgrade the utility of, and develop intelligence in, a computer network by offering the means to create and manipulate information containers with dynamic registers.

2. Description of the Related Art

In the present day, querying and usage of information resources on a computer network is accomplished by individuals directing a search effort by submitting key words or phrases to be compared to those key words or phrases contained in the content or description of that information resource, with indices and contents residing in a fixed location unchanging except by human input. Similarly, the class of storage medium upon which information resides, it class and subclass organizational structures, and its routes of access all remain fundamentally unaltered by ongoing user queries and usage. Only the direct and intended intervention of the owner of the information content or computer hosting site changes these parameters, normally accomplished manually by programmers or systems operators at their own discretion or the discretion of the site owner.

There exists currently in the art a limited means of interfacing a computer user with the information available on computer networks such as the world wide web. Primarily, these means are search engines. Search engines query thousands or tens of thousands of index pages per second to suggest the location of information while the user waits. While factual information can be accessed, the more complex, particular or subtle the inquiry, the more branches and sub-branches need to be explored in a time consuming fashion in order to have any chance of success. Further, there are no such automatic devices that reconstruct the information into more useful groupings or makes it more accessible according to factors attached to the content by the content creator such as the space or time relevancy of its content, or factors attached to the content by the system's compilation and analysis of the accumulated biography of that specific content's readership.

The utility of wide area and public computer networks is thus greatly limited by the static information model and infrastructure upon which those networks operate.

One problem is that on a wide area or public network, specific content such as a document remains inert, except by the direct intervention of users, and is modified neither by patterns or history of usage on the network, or the existence of other content on the network.

Another problem is that content does not reside in an information infrastructure conducive to reconstruction by expert rule-based, fuzzy logic, or artificial intelligence based systems. Neither the intelligence of other information users nor the expert intelligence of an observant network computer system can be utilized in constructing, or re-constructing information resources. Where content resides in a fixed location and structure, "information" becomes something defined by the mind of the information provider rather than the mind of the information user, where the actual construction and utility of information exists. Information remains, like raw ore, in an unrefined state.

Another problem is that the class of storage medium upon which data resides cannot be system or user managed and altered according to the actual recorded and analyzed hierarchically graded usage of any given information resource residing on that storage medium except by statistical analysis of universal, undefined "hits" or visits to that page or site.

Another problem is that information resource groupings remain fixed on the given storage medium location according to the original installation by the resource author, not altered according to the actual recorded and analyzed hierarchically graded usage of that given information resource. Content itself remains inert, with no possibility of evolution.

A further problem with the prior art is that neither the search templates generated by those more knowledgeable in a given field of inquiry, nor the search strategies historically determined to be successful, or system-constructed according to analyses of search strategies historically determined to be successful, are available to inquiring users. A search template is here defined as one or more text phrases, graphics, video or audio bits, alone or in any defined outline or relational format designed to accomplish an inquiry. Internet or wide area network search may return dozens of briefs to a keyword or key phrase inquiry sometimes requiring the

time-consuming examination of multiple information resources or locations, with no historical relation to the success of any given search strategy.

A further problem is that there is limited means to add to, subtract from, or alter the information content of documents, databases, or sites without communicating with the owners or operators of those information resources, e.g., contacting, obtaining permission, negotiating and manually altering, adding or subtracting content. Additionally, once so altered, there is not a means to derive a proportionate value, and thereby a proportionate royalty as the information is used.

A final problem is that the physical residence of a body of data or its cyberspace location may not serve its largest body of users in the most expedient manner of access. Neither the expert intelligence of other information users nor the expert intelligence of an observant computer system is presently utilized by inherent network intelligence to analyze, re-design and construct access routes to information medium except by statistical analysis of universal, undefined "hits" or visits to that page or site.

Therefore, there is a need for a system and methods for creating and manipulating information containers with dynamic interactive registers defining more comprehensive information about contained content in a computer, media or publishing network, in order to manufacture information on, upgrade the utility of, and develop intelligence in, a computer network by providing a searching user the means to utilize the searches of other users or the historically determined and compiled searches of the system, a means to containerize information with multiple registers governing the interaction of that container, a means to re-classify the storage medium and location of information resources resident on the network, a means to allow the reconstruction of content into more useful formations, and a means to reconstruct the access routes to that information.

SUMMARY OF THE INVENTION

The present invention is a system and methods for manufacturing information on, upgrading the utility of, and developing intelligence in, a computer or digital network, local, wide area, public, corporate, or digital-based, supported, or enhanced physical media form or public or published media, or other by offering the means to create and manipulate information containers with dynamic registers.

The system of the present invention comprises an input device, an output device, a processor, a memory unit, a data storage device, and a means of communicating with other computers, network of computers, or digital-based, supported or enhanced physical media forms or public or published media. These components are preferably coupled by a bus and configured for multi-media presentation, but may also be distributed throughout a network according to the requirements of highest and best use.

The memory unit advantageously includes an information container made interactive with dynamic registers, a container editor, a search interface, a search engine, a search engine editor, system-wide hierarchical container gateways interacting with dynamic container registers, a gateway editor, a register editor, a data collection means with editor, a data reporting means with editor, an analysis engine with editor, an executing engine with editor, databases, and a means of communicating with other computers as above. These components may reside in a distributed fashion in any configuration on multiple computer systems or networks.

The present invention advantageously provides a container editor for creating containers, containerizing storing information in containers and defining and altering container registers. A container is an interactive nestable logical domain configurable as both subset and superset, including a minimum set of attributes coded into dynamic interactive evolving registers, containing any information component, digital code, file, search string, set, database, network, event or process, and maintaining a unique network-wide lifelong identity.

The container editor allows the authoring user to create containers and encapsulate any information component in a container with registers, establishing a unique network lifelong identity, characteristics, and parameters and rules of interaction. The authoring user defines and sets the register with a starting counter and/or mathematical description by utilizing menus and simple graphing tools or other tools appropriate to that particular register. The registers determine the interaction of that container with other containers, system components, system gateways, events and processes on the computer network.

Containers and registers, upon creation, may be universal or class-specific. The editor provides the means to create system-defined registers as well as the means to create other registers. The editor enables the register values to be set by the user or by the system, in which case the register value may be fixed or alterable by the user upon creation. Register values are

evolving or non-evolving for the duration of the life of the container on the system. Evolving registers may change through time, space, interaction, system history and other means.

System-defined registers comprise: (1) an historical container register, logging the history of the interaction of that container with other containers, events and processes on the network, (2) an historical system register, logging the history of pertinent critical and processes on the network, (3) a point register accumulating points based upon a hierarchically rated history of usage, (4) an identity register maintaining a unique network wide identification and access location for a given container, (5) a brokerage register maintaining a record of ownership percentage and economic values, and others.

The present invention also includes user-defined registers. User defined registers may be created wholly by the user and assigned a starting value, or simply assigned value by the user when that register is pre-existent in the system or acquired from another user, and then appended to any information container, or detached from any container.

Exemplary user-defined registers comprise (1) a report register, setting trigger levels for report sequences, content determination and delivery target, (2) a triple time register, consisting of a range, map, graph, list, curve or other representation designating time relevance, actively, assigning the time characteristics by which that container will act upon another container or process, passively, assigning the time characteristics by which that container be acted upon by another container or process, and neutrally, assigning the time characteristics by which that container will interact with another container or process, (3) a triple space register, consisting of a range, map, graph, list, curve or other representation designating the domain and determinants of space relevance, actively, assigning the space characteristics by which that content will act upon another container or process, passively, assigning the space, characteristics by which that content will be acted upon by another container or process, and neutrally, assigning the space characteristics by which that container will interact with another container or process, (4) a domain of influence register, determining the set, class and range of containers upon which that container will act, (5) a domain of receptivity register, determining the set, class and range of containers allowed to act upon that container, (6) a domain of neutrality register, determining the set, class and range of containers with which that container will interact, (7) a domain of containment register, determining the set, class and range of containers which that container may logically encompass, (8) a domain of inclusion register, determining the set, class and range of containers by which that container might be encompassed, (9) an ownership register, recording the original ownership of that containers, (10) a proportionate ownership register, determining the proportionate ownership of that containers, (11) a creator profile register, describing the creator or creators of that container, (12) an ownership address register, maintaining the address of the creator or creators of that container, (13) a value register, assigning a monetary or credit value to that container, and (14) other registers created by users or the system.

Containers are nestable and configurable as both subset and superset and may be designated hierarchically according to inclusive range, such as image component, image, image file, image collection, image database, or if text, text fragment, sentence, paragraph, page, document, document collection, document, database, document library, or any arrangement wherein containers are defined as increasingly inclusive sets of sets of digital components.

The present invention also includes, structurally integrated into each container, or strategically placed within a network at container transit points, unique gateways, nestable in a hierarchical or set and class network scheme. Gateways gather and store container register information according to system-defined, system-generated, or user determined rules as containers exit and enter one another, governing how containers system processes or system components interact within the domain of that container, or after exiting and entering that container, and governing how containers, system components and system processes interact with that unique gateway, including how data collection and reporting is managed at that gateway. The gateways record the register information of internally nested sub and superset containers, transient containers and search templates, including the grade of access requested, and, acting as an agent of an analysis engine and execution engine, govern the traffic and interaction of those containers and searches with the information resource of which they are the gateway and other gateways. The gateways' record of internally nested and transient container registers, and its own interaction with those containers, is made available, according to a rules-based determination, to the process of the analysis engine by the data collection and/or data reporting means.

The present invention also includes a means of data storage at any given gateway.

The present invention also includes a data collection means, residing anywhere on the network, or located at one or more hierarchical levels of nestable container gateways for gathering information from other gateways and analysis engines according to system, system-generated or user determined rules. The data collection means manages the gathering of data regarding network-wide user choices, usage and information about information, by collecting it from container and gateway registers as those containers and gateways pass through one another. Such statistics as frequency, pattern, and range of time, space and logical class is collected as directed by the analysis engine, and made that data available to the analysis engine by advancing it directly to the analysis engine, or incrementally, to the next greater hierarchically inclusive collection level. The rules of data collection may be manually set or altered by the system manager, or set by the system and altered by the system in its evolutionary capacity.

The present invention also includes a data reporting means, located at one or more hierarchical levels of nestable container gateways for submitting information to other gateways and analysis engines according to system, system-generated or user determined rules. The data reporting means manages the sending of data from the registers, gateways and search templates in a frequency, pattern, and range of time, space and logical class as directed by the analysis engine, and makes that data available to the analysis engine by advancing it directly to the analysis engine, or incrementally to the next greater hierarchically inclusive reporting level. The rules of data collection may be manually set or altered by the system manager, or set by the system and altered by the system in its evolutionary capacity. The data reporting means may be established to work in concert, in redundancy, or in contiguous or interwoven threads of hierarchically nested containers.

The present invention also includes an analysis engine that receives, reports and collects information regarding the interaction of user searches with gateways and container registers, as well as container registers with other container registers, and container registers with gateways. The analysis engine analyzes the information submitted by the gateways and instructs the execution engine to create new information containers, content assemblages, storage schemes, access routes, search templates, and gateway instructions. The analysis engine includes an editor that provides a system manager with a means of editing the operating principles of that engine, governing data reporting, data collection, search template loading, gateway instructions, and other.

The present invention also includes an execution engine, fulfilling the instructions of the analysis engine, to create new information containers, content sun and superset assemblages, storage schemes, access routes, search templates, and gateway instructions. The execution engine includes an editor that provides a system manager with a means of editing the operating principles of that engine, governing data reporting, data collection, search template loading, gateway instructions, and other.

The present invention also includes a search interface or browser. The search interface provides a means for a searching user to submit, record and access search streams or phrases generated historically by himself, other users, or the system. Search streams or phrases of other users are those that have been historically determined by the system to have the highest probability of utility to the searching user. Search streams or phrases generated by the system are those that have been constructed by the system through the analysis engine based upon the same criteria.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a first and preferred embodiment of a system constructed according to the present invention.

FIG. **2 A** is block diagram of a preferred embodiment of the memory unit.

FIG. **2 B** is an exemplary embodiment of a computer network showing computer servers, personal computers, workstations, Internet, Wide Area Networks, Intranets in relationship with containers and gateways.

FIG. **2B1** is an exemplary embodiment of a computer network showing computer servers, personal computers, workstations, Internet, Wide Area Networks, Intranets in relationship with containers and gateways and exemplary locations of gateway storage in proximity to one or more of the various sites.

FIGS. **2C** through **2H** are exemplary embodiments in block diagram form of computer network components showing a possible placement of nested containers, computer servers, gateways, and the software components named in Fig. 2 A on a network.

FIG. **3A** is a graphical representation for one embodiment of a container having a plurality of containers nested within that container.

FIG. **3C** is a drawing showing elements that might be logically encapsulated by a container. FIG. **4** is a drawing of an information container showing a gateway and registers logically

encapsulating containerized elements.

FIG. **5** is a flowchart showing a preferred method for the containerization process and container editor operating on the communication device.

FIG. **6** is a flowchart showing a preferred method for searching for containers within a node.

FIG. **7** is a flowchart further showing a preferred method for searching for containers over one or more gateways.

FIG. **8** is a flowchart showing a method for performing the data collection and reporting on containers.

FIG. **9** is a flowchart showing the operation of the analysis engine.

FIG. **10** is a flowchart showing the operation of the execution engine.

FIG. **11** is a flowchart showing the operation of the gateway editor.

FIG. **12** is a flowchart showing the operation of the gateway process.

FIG. **13A** is a drawing showing an example of nested containers, gateways, registers, analysis engines and an execution engine prior to container reconstruction as depicted in 13 B, 13 C and 13 D.

FIG. **13B** is a drawing showing the reconstructed nested containers of Figure 13A.

FIG. **13C** is a drawing showing further reconstruction of nested containers, with a container relocated to reside within another container.

FIG. **13D** is a drawing showing a flowchart of the reconstruction process

FIG. **14** is a drawing showing the screen interface of the container editor.

FIG. **15** is a drawing showing the screen interface of the gateway editor.

FIG. **16** is a drawing showing the screen interface of the search interface.

FIG. **17** is a drawing of a generic application program showing a drop-down menu link, and a button link to the containerization process or container editor.

DESCRIPTION OF THE PREFERRED EMBODIMENT

THE SYSTEM

Referring now to FIG. **1**, a preferred embodiment of a system **10** for creating and manipulating information containers with dynamic interactive registers in a computer, media, or publishing network **201** in order to manufacture information on, upgrade the utility of, and develop intelligence in that network **201**, is shown. The system **10** preferably comprises an input device **24**, an output device **16**, a processor **18**, a memory unit **22**, a data storage device **20**, and a communication device **26** operating on a network **201**. The input device **24**, an output device **16**, a processor **18**, a memory unit **22**, a data storage device **20**, are preferably coupled together by a bus **12** in a von Neumann architecture. Those skilled in the art will realize that these components **24**, **16**, **18**, **22**, **20**, and **26** may be coupled together according to various other computer architectures including any physical distribution of components linked together by the communication device **26** without departing from the spirit or scope of the present invention, and may be infinitely nested or chained, both as computer systems within a network **202**, and as networks within networks **201**.

The output device **16** preferably comprises a computer monitor for displaying high-resolution graphics and speakers for outputting high fidelity audio signals. The output device **16** is used to display various user interfaces **110**, **125**, **210**, **300**, **510**, **610**, **710**, as will be described below, for searching for and containerizing information, and editing the container gateways, containers, container registers, the data reporting means and the data collection means, and the search, analysis and execution engines. The author uses the input device **24** to manipulate icons, text, charts or graphs, or to select objects or text, in the process of packaging, searching or editing in a conventional manner such as in the Macintosh of Windows operating systems.

The processor **18** preferably executes programmed instruction steps, generates commands, stores data and analyzes data configurations according to programmed instruction steps that are stored in the memory unit **22** and in the data storage device **20**. The processor **22** is preferably a microprocessor such as the Motorola 680(x)0, the Intel 80(x)86 or Pentium, Pentium II, and successors, or processors made by AMD, or Cyrix CPU of the any class.

The memory unit **22** is preferably a predetermined amount of dynamic random access memory, a read-only memory, or both. The memory unit **22** stores data, operating systems, and programmed instructions steps, and manages the operations of all hardware and software components in the system **10** and on the network **201**, utilizing the communication device **26** whenever necessary or expeditious to link multiple computer systems **202** within the network **201**.

The data storage device **20** is preferably a disk storage device for storing data and programmed instruction steps. In the exemplary embodiment, the data storage device **20** is a hard disk drive. Historical recordings of network usage are stored on distributed and centralized data storage devices **20**.

The preferred embodiment of the input device **24** comprises a keyboard, microphone, and mouse type controller. Data and commands to the system **10** are input through the input device **24**.

The present invention also includes a communication device **26**. The communication device **26** underlies and sustains the operations of, referring now also to Fig **2** the analysis **400** and execution **500** engines, the data reporting **600** and collection **700** means, the container editor **110,** the search interface **300,** and the search engine **320,** providing the means to search, access, move, copy, utilize or otherwise perform operations with and on data. The communication device **26** utilizes one or more of the following technologies: modem, infrared, microwave, laser, photons, electrons, wave phenomena, cellular carrier, satellite, laser, router hub, direct cabling, physical transport, radio, broadcast or cable TV or other to communicate with other computers, digital-supported television, computer networks, or digital-based or supported public or published media, or physical media forms, on any a local, wide area, public, or any computer-based computer supported, or computer interfaced network, including but not limited to the Internet. It also allows for the functioning and distribution of any container **100** or container component herein described to reside anywhere on any computer system in any configuration on that local, wide area, public, or corporate computer-based or computer related network, or digital-based or supported media form.

Referring now to Figure **2 A**, a preferred embodiment of the memory unit **22** is shown. The memory unit includes: an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510**, a search interface **300**, search engine **320**, analysis engine **400**, execution engine **500**, a data reporting module, **600**, a data reporting editor **610**, a data collection module **700**, a data collection editor **710**, screen interfaces (GUI's) **936**, menu or access buttons from generic computer programs **937**, and databases **900**, all residing in memory optimized between a data storage means **20** such as magnetic, optical, laser, or other fixed storage, and a memory means **22** such as RAM. The memory unit **22** functions by operating on communications network **12** with a communication device **26** on multiple computer systems **202** within the network **201**. These components will be described first briefly in the following paragraphs, then in more detail with reference to Figures 3 A through 17.

Those skilled in the art will realize that these components might also be stored in contiguous blocks of memory, and that software components or portions thereof may reside in the memory unit **22** or the data storage means **20.**

The present invention includes information containers **100** as noted above. The information container **100** is a logically defined data enclosure which encapsulates any element or digital segment (text, graphic, photograph, audio, video, or other), or set of digital segments, or referring now to FIG. **3 C,** any system component or process, or other containers or sets of containers. A container **100** at minimum includes in its construction a logically encapsulated portion of cyberspace, a register and a gateway. A container **100** at minimum encapsulates a single digital bit, a single natural number or the logical description of another container, and at maximum all defined cyberspace, existing, growing and to be discovered, including but not limited to all containers, defined and to be defined in cyberspace. A container **100** contains the code to enable it to interact with the components enumerated in 2 A, and to reconstruct itself internally and manage itself on the network **201.**

The container **100** also includes container registers **120**. Container registers **120** are interactive dynamic values appended to the logical enclosure of an information container **100,** and serve to govern the interaction of that container **100** with other containers **100,** container gateways **200** and the system **10,** and to record the historical interaction of that container **100** on the system **10.** Container registers **120** may be values alone or contain code to establish certain parameters in interaction with other containers **100** or gateways **200.**

The present invention also includes container gateways **200.** Container gateways **200** are logically defined gateways residing both on containers **100** and independently in the system **10.** Gateways **200** govern the interactions of containers **100** within their domain, and alter the registers **120** of transiting containers **100** upon ingress and egress.

The present invention also includes container gateway storage **205** to hold the data collected from registers **120** of transient containers **100** in order to make it available to the data collection means **700** and the data reporting means **600**, and to store the rules governing the operations of its particular gateway **200**, governing transiting containers upon ingress and egress, and governing the interactive behavior of containers **100** within the container **100** to which that gateway **200** is attached. Gateway storage **205** may be located on gateways **200** themselves, containers **100** or anywhere on the network **202**, **201**, including but not limited to Internet, Intranet, LAN, WAN, according to best analysis and use.

The memory unit **22** also includes an execution engine **500** to perform the functions on the system **10** as directed by the analysis engine after its analysis of data from the data reporting means **600**, the data collection means **700**, and the search interface **300**.

The memory unit **22** also includes a search interface **300**, by which the user enters, selects or edits search phrases or digital strings to be used by the search engine **320** to locate containers **100**.

The memory unit **22** also includes an analysis engine **400** which performs rules based or other analysis upon the data collected from the search interface **300** and the data collection **700** and data reporting **600** means.

The memory unit **22** also includes a data reporting means **600**, by which means the information collected by gateways **200** from transient containers **100** is sent to the analysis engine **400**.

The memory unit **22** also includes a data collection means **700**, by which means the analysis engine **400** gathers the information collected by gateways **200** from transient containers **100**.

The memory unit **22** also includes a container editor **110** for creating, selecting, acquiring, modifying and appending registers **120** and gateways **200** to containers **100**, for creating, selecting, acquiring, and modifying containers, and for selecting content **01** to encapsulate.

The memory unit **22** also includes a register editor **125**, for creating, selecting, acquiring and modifying container registers **120** and establishing and adjusting the values therein.

The memory unit **22** also includes a gateway editor **210**, by which means the user determines the rules governing the interaction of a given gateway **210** with the registers **120** of transient containers **100,** governing transiting containers upon ingress and egress, and governing the interactive behavior of containers within the container to which that gateway is attached.

The memory unit **22** also includes databases **900,** by which means the analysis engine **400,** the execution engine **500**, the gateways **100**, the editors **110**, **125**, **210**, **510**, **610**, **710**, and the search interface **300**, store information for later use.

The memory unit **22** present invention also includes a search engine **320** by which means the user is able to locate containers **100** and, referring now to Fig. **4**, containerized elements **01**.

The memory unit **22** present invention also includes an engine editor **510,** by which means the user establishes the rules and operating procedures for the analysis engine **400** and the execution engine **500**.

The memory unit **22** present invention also includes a reporting means editor **610**, by which means the user establishes the rules and schedule under which the information collected by gateways **200** from transient containers **100** will be sent to the analysis engine **400**.

The memory unit **22** present invention also includes a collection means editor **710,** by which means the user establishes the rules and schedule under which the analysis engine **400** will gathers the information collected by gateways **200** from transient containers **100**.

The memory unit **22** present invention also includes screen interfaces (GUI's) **936**, specifically designed to simplify and enhance the operations of the container editor **110**, the gateway editor **210**, and the search interface **300**.

The present invention also includes a menu or button access **937**, by which a user utilizing any generic computer program may access the system **10** or the container editor **110** from a menu selection(s) or button(s) within that program.

The present invention also includes a computer, media or publishing network **201**, comprising computers, digital devices and digital media **202** and a communication device **26**, within which the components enumerated in Fig. **2 A** interact, compiling, analyzing, and altering containers **100** and the network **201**according to information gathered from container registers **120**.

The memory unit **22** also includes one or more computers **202**, by which means the components of Fig **1** sustain the operations described in Fig. **2 A.**

The memory unit **22** also includes flat or relational databases **900**, used where, and as required. Databases are used to store search phrases, search templates, system history for the analysis engine and execution engine, container levels and container, sites and digital elements, or any and all storage required to operate the system.

Referring now to FIG. **2 B,** a drawing of a computer network **201** as a system **10,** showing a possible placement of nested containers **100**, computer servers, gateways **200**, on the sites described below. (Note: Fig. 2 B utilizes in parts the same numbering scheme as Fig. 13 A, 13 B, 13 C, 13 D and as Fig. 2 A.) In FIG. **2 B** various exemplary sites are shown, any or all of which might interact dynamically within the system. Site 1 shows a single workstation with a container and gateway connected to an Intranet. .(Individual containers may be a floppy or CD-Rom to be downloaded or inserted.) Site 2 shows a server with a gateway in relationship to various containers.. Site 3 shows an Internet web page with a container residing on it. Site 4 shows a personal computer with containers and a gateway connected to the Internet. Site 5 shows a configuration of multiple servers and containers on a Wide Area Network.. Site 6 shows a workstations with a gateway and containers within a container connected to a Wide Area Network. Site 7 shows an independent gateway, capable of acting as a data collection and data reporting site as it gathers data from the registers of transiting containers, and as an agent of the execution engine as it alters the registers of transient containers. A container **100** contains the code to enable it to interact with the components enumerated in **2A,** and to reconstruct itself internally and manage itself on the network **201.** The code resides in and with the container in its registers and gateway definitions and controls. Additional system code resides in all sites to manage the individual and collective operation and oversight of the components enumerated in **2A**, with the specific components distributed amongst the sites according to the requirements of optimization.

Referring now to Fig. **2 B 1** various exemplary sites are shown as described above in Fig. **2 B**, with the addition of possible location of one or more gateway storage **205** locations.

Referring now to Figures **2 C** through **2 H**, various exemplary sites with one or more of the logical components of the system 10 in relationship are shown. Site 1 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510**, a search interface **300**, search engine **320**, analysis engine **400**, execution engine **500**, a data reporting means **600**, a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 2 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510,** search engine **320**, analysis engine **400**, execution engine **500**, a data reporting means **600**, a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20,** utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 3 comprises an interactive information container 100, a container editor **110**, container registers **120**, a container register editor **125**, hierarchical container gateways 200, gateway storage 205, gateway editors **210**, and databases **900**, all residing on data storage means **20,** utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 4 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, a search interface **300**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 5 comprises an interactive information container **100**, container registers **120**, a container register editor **125,** hierarchical container gateways **200**, gateway storage **205,** and databases **900**, all residing on data storage means **20**, accessed and utilized by non-resident memory unit **22**, operating on communications network **12** with a communication device **26.**

Site 6 includes an independent analysis engine **400,** execution engine **500**, data collection means **700**, and data reporting means **600** gateway editors **210**, engine editors **510,** a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Referring now to FIG. **3 A** and FIG. **3 B**, a block diagram of several nested information containers is shown, including examples of elements, e.g., code **1100**, text **1200**, audio **1300**, video **1400**, photograph **1500**, graphic images **1600**, and examples of possible container level classifications in increasing size, e.g., element **10900000**, document **10800000**, database **10700000**, warehouse **10600000**, domain **10500000**,and continuing increasingly larger on Fig **3 (B)**, subject **10400000**, field **10300000**, master field **10200000**, species **10100000**. Containers may be infinitely nested and assigned any class, super class or sub class scheme and description by the creator of the container to govern nesting within that container. In addition to digital elements, containers may also include system process and components, including containerization itself.

Referring now to FIG. **3 C**, a block diagram of an information container system is shown, listing, without any relationship indicated, some of the possible system components and processes, or sets thereof, that may be encapsulated as elements **01** in an information container **100.** An information container 100 may include one or more of the following: any unique, container **100**, gateway **200**, output device **16**, input device **24**, output device process **160**, input device process **240**, data storage device **20**, data storage device process **2000**, processor **18**, bus **12**, content **01**, search process **02**, interface **04**, memory unit **22**, communication device **26**, search interface **300**, search process **98**, network **201**, class of device, process or content **999**, class of process at any unique class of device **990**, process at any unique device **99**, editor **110**, **125**, **210**, **510**, **610**, **710**, engine **320**, **400**, **500**, containerization process **1098**, or process **08**.

Any container may include (n) other containers, to infinity. The use of value evolving container registers **120** in conjunction with gateways **200**, data reporting modules **600**, data collection modules **700**, the analysis engine **400**, and the execution engine **500** provides the information container **100** with extensive knowledge of the use, operation of its internal contents, prior to, during and after those contents' residence within that container **100**, and extensive knowledge of the use, operation and contents of the system **10** external to itself, and allows the container **100** to establish and evolve its own identity and course of interaction on the system **10**. Further, containers **100**, as logical enclosures, can exist and operate independent of their digital contents, whether encapsulating audio, video, text, graphic, or other.

Referring now to FIG. **4**, a block diagram of an information container **100** is shown. The information container **100** is a logically defined data enclosure which encapsulates any element, digital segment (text, graphic, photograph, audio, video, or other), set of digital segments as described above with reference to FIG. **3 (C)**, any system component or process, or other containers or sets of containers. The container **100** comprises the containerized elements **01**, registers **120** and a gateway **200**.

Registers **120** appended to an information container **110** are unique in that they operate independently of the encapsulated contents, providing rules of interaction, history of interaction, identity and interactive life to that container **100** through the duration of its existence on a network **201**, without requiring reference to, or interaction with, its specific contents. They enable a container **100** to establish an identity independent of its contents. Additionally, registers **120** are unique in that their internal values evolve through interaction with other containers **100**, gateways **200**, the analysis engine **400**, the execution engine **500**, and the choices made by the users in the search interface **300**, the container editor **110**, the register editor **125**, the gateway editor **210**, the engine editor **510**. Registers **120** are also unique in that they can interact with any register of a similar definition on any container **100** residing on the network **201**, independent of that container's contents. Registers **120**, once constructed, may be copied and appended to other containers **100** with their internal values reset, to form new containers. Register values, when collected at gateways **200** and made available to the analysis engine **400** through the data collection means **700** and the data reporting means **600**, provide an entirely new layer of network observation and analysis and operational control through the execution engine **500**. Registers **120** accomplish not only a real time information about information system, but also a real time information about information usage on a network. Further, because the user base of a network determines usage, the system **10**, in gathering information about information usage, is observing the choices of the human mind. When these choices are submitted to the analysis of a rules-based or other analysis engine **400**, the system **10** becomes capable of becoming progressively more responsive to the need of the user base, in effect, learning to become more useful by utilizing the execution engine **500** to create system-wide changes by altering the rules of gateway **200** interaction and thereby altering the registers **120** of transient containers **100** and establishing a complete evolutionary cycle of enhanced utility.

Further, in establishing the pre-defined registers as described in the following four paragraphs, the following unique aspects of information about information are utilized for the first time: 1) the dynamic governance of information according to its utility through time, in active, passive and neutral aspects, as explained below; 2) the dynamic governance of information according to its utility through space in active, passive and neutral aspects, as explained below; 3) the dynamic governance of information according to its ownership, as explained below; 4) the dynamic governance of information according to its unique history of interaction as an identity on a network, as explained below; 5) the dynamic governance of information according to the history of the system on which it exists, as explained below; 6) the dynamic governance of information according to established rules of interaction, in active, passive and neutral aspects, as explained below; 7) the dynamic governance of information according to the profile of its creator, as explained below; 8) the dynamic governance of information according to the value established by its ongoing usage, as explained below; 9) the dynamic governance of information according to it distributed ownership, as explained below; 10) the dynamic governance of information according to what class of information it might be incorporated into, and according to what class of information container it might incorporate, as explained below; 11) the dynamic governance of information according to self-reporting, as explained below.

Referring now to Fig **4**, registers **120** may be (1) pre-defined, (2) created by the user or acquired by the user, or (3) system-defined or system-created. Pre-defined registers **120** are those immediately available for selection by the user within a given container editor as part of that container editor, in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. Registers **120** created by the user are those conceived and created by a specific user or user group and made immediately available for selection by the user or user group in conjunction with any of a wide number of container editors, in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. Registers **120** acquired by the user are those registers existing network-wide **201**, created by the user base, that might be located and acquired by the user in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. System-defined registers are those registers whose values are set and/or controlled by the system **10**. System-created registers are those registers created by the system **10**.

Registers **120** are user or user-base created or system-created values or ranges made available by the system **10** to attach to a unique container, and hold system-set, user-set, or system-evolved values. Values may be numeric, may describe domains of time or space, or may provide information about the container **100**, the user, or the system **10**. Registers **120** may be active, passive or interactive and may evolve with system use. Pre-defined registers include, but are not limited to, system history **110000**, container history **101000**, active time **102000**, passive time **103000**, neutral time **104000**, active space **111000**, passive space **112000**, neutral space **113000**, containment **105000**, inclusion **106000**, identity **114000**, value **115000**, ownership **107000**, ownership addresses **116000**, proportionate ownership **117000**, creator profile **108000**, receptivity **118000**, influence **119000**, points **109000**, others **120000**, reporting **121000**, neutrality **122000**, acquire **123000**, create **124000**, content title **125000**, content key phrase(s) **126000**, and content description **127000,** security **12800**, and parent rules **129000**.

Pre-defined registers comprise an historical container register **101000**, logging the history of the interaction of that container **100** with other containers, events and processes on the network **201**, an historical system register **110000**, logging the history of pertinent critical and processes on the network, a point register **109000** accumulating points based upon a hierarchically rated history of usage, an identity register **114000** maintaining a unique network wide identification and access location for a given container specifying a unique time and place of origin and original residence, a proportionate ownership register **117000** maintaining a record of ownership percentage and economic values, and others **120000**.

User-defined registers include a report register **121000** setting trigger levels for report sequences, content determination and delivery target, three time registers, consisting of a range, map, graph, list, curve or other designating time relevance, **102000** assigning the time characteristics by which that container will act upon another container or process, **103000** assigning the time characteristics by which that container be acted upon by another container or process, and **104000** assigning the time characteristics by which that container will interact with another container or process, three space registers, consisting of a range, map, graph, list, curve or other designating the domain and determinants of space relevance, **111000** assigning the space characteristics by which that content will act upon another container or process, **112000** assigning the space, characteristics by which that content will be acted upon by another container or process, and **113000** assigning the space characteristics by which that container will interact with another container or process, a domain of influence register **119000**, determining the set, class and range of containers upon which that container will act, a domain of receptivity register **118000**, determining the set, class and range of containers allowed to act upon that container, a domain of neutrality register **122000**, determining the set, class and range of containers with which that container will interact, a domain of containment register **105000**, determining the set, class and range of containers which that container may logically encompass, a domain of inclusion **106000** register, determining the set, class and range of containers by which that container might be encapsulated, an ownership register **107000**, recording the original ownership of that containers, a creator profile register **108000**, describing the creator or creators of that container, an ownership address register **116000**, maintaining the address of the creator or creators of that container, a value register **115000**, assigning a monetary or credit value to that container, other registers **120000** created by users or the system, a reporting register **121000**, determining the content, scheduling and recipients of information about that container, a neutrality register **122000**, an acquire register **123000**, enabling the user to search and utilize other registers residing on the network, a create register **124000**, enabling the user to construct a new register, a content title register **125000**, naming the contents of the container, a content key register, **126000**, identifying the container contents with a key phrase generated by the user and/or the system based upon successful usage of that phrase in conjunction with the utilization of the information within that container **100**, a content description register **127000**, identifying the container contents with additional description, a security register **128000**, controlling container security, and a parent container register **129000**, storing the rules governing container interaction as dictated by the parent (encapsulating) container.

The container also includes a gateway **200** and gateway storage **205.**

Gateways **200** are logically defined passageways residing both on containers **100** and independently in the system **10**. Gateways **200** govern the interactions of containers **100** encapsulated within their domain by reading and storing register **120** information of containers entering and exiting that container **100.**

The present invention also includes container gateway storage **205.** Gateway storage **205** stores information regarding the residence, absence, transience, and alteration of encapsulated and encapsulating containers **100**, and their attached registers **120, holding** the data collected from registers **120** of transient containers **100** in order to make it available to the data collection means **700** and the data reporting means **600**, and storing the rules governing the operations of its particular gateway **200.**

Referring now to FIG. **5**, a flow chart of the preferred method for creating a container **100**.is shown.

Input is received from the user selecting a container level through use of a drop-down menu **10100**. A menu of all possible container classes within the subset and superset scheme of multiple hierarchically nested containers, i.e.; element, document, file, database, warehouse, domain, and more, is displayed on the output device **10200.** Input is received from the user selecting a class **10300.**

A graphic representation of a container in that class, with registers common to all containers as well as registers unique to its class is displayed **10301.**

Input is received from the user choosing to "create" **10400**, "edit" **10500**, or "locate" **10600**.

When the input of "create" **10400** is received from the user, a container template in that class appears **10410**. Input from the user is then received adding or selecting a register 10540 to append to that container template. When input is received from the user adding a register, a list of registers that might be added to that class of container is made available to select **10550.** Input is received from the user selecting a register **10560** and editing it **10570**. The menu returns to "add or select" **10540**.

If the input of "locate" **10600** is received from the user, the system prompts the user to enter the identity of the container or class of containers **10605**. The system locates the container(s) **10610**. Input is received from the user selecting a container **10620**. The system prompts the user for a security code for permission to access the container for template use, or to alter its registers, or to alter its content **10630**. . Input is received from the user entering a name and password providing access to one of the security levels **10640.** Input is received from the user editing the container accordingly by transition to step **10500** and performing the steps for editing.

If the input of "edit" **10500** is received, a list of containers available to edit at that level is shown **10510**. Input is received from the user selecting a container **10520**. That container appears, available to edit **10530**. Input is received from the user selecting "add" or "select" registers **10540** by the user clicking on the graphically depicted register, or from a drop down menu. Input is received from the user selecting the register to edit **10560**. Input is received from the user selecting "modify" or "delete" for that register **10565**. If input is received from the user to "delete," that register is severed from the container. If input is received from the user to "modify", the register editor **10570** screen appropriate to that register appears, i.e., an x-y type graph to define a curve of relevant active time, in which the user manipulates the x-y termini, scale and curve, or a global map in which Input is received from the user selecting the locale of active space, whether zip code, city, county, state, country, continent, plant or other. When input is received from the user saving the definition, the screen returns to the main container screen to make another selection available. . Input is received from the user defining as many registers as he chooses. One of the registers may be named "new register." Input is received from the user selecting the new register, and if chosen by the user, defining a wholly unique and new kind of register by the user entering input into the register editor **125.**

When the input is received from the user choosing to add a register, a list of registers that might be added to that class of container are made available to select **10550**. Input is received from the user selecting a register **10560** and editing it **10570**. The menu returns to "add or select" **10540**, and in turn to Input – Select Container.

Input may then be received from the user choosing to add, modify, or delete the container contents **10700**. Once the registers are defined, input is received from the user indicating completion and the interface reverts to the container editor. When input is received from the user choosing "select component" (to select the component to containerize) from the main menu bar **10700, a** window appears allowing the user to select any file, component, or other container. If for example, the user were creating a warehouse container, and wishes to incorporate several databases into that container, input would then be received from the user selecting "database." The program would prompt the user for the location (directory) of that database or container. If the requested selection is not containerized, input may then be received from the user choosing to containerize the element at that time, after which the program returns to "select component." Once input is received from the user defining the database location, the program logically encases the directory or directories in the defined container. The above procedure may be repeated as many times as desired to include multiple databases within a single container. While logical simplicity would dictate that all containers within a container be of the same subset, it would be possible for input to be received from the user choosing containers of any subset to include in the container. When input is received from the user choosing "finished," the container is created with a unique network identity, preferably through some combination of exact time and digital device serial number, or centralized numbering system, or other means. The container **100** contains all digital code, including data and program software from the selected items or containers.

Input may then be received from the user to publish the container **11100** at a user-identified or system suggested location **11200** to be selected **11400**.

Input is received from the user to "publish", from the main menu bar **11100**. Input is received from the user choosing to leave the container where it was created, move or copy it to another drive, directory, computer, or network the user designates, or select the location from location options offered by the system **11200**, or submit, or duplicate and submit, the container to the analysis engine **400** for intelligent inclusion in other containers, thus allowing the system to publish the container as instructed or choose the residence of the container **11400**.

If input is received from the user to choosing to "move," or "copy" a browse function allows the user to name the new location or browse a list of possible locations. If input is received from the user choosing to "submit," a browser function allows the user to name the analysis search engine **310** or browse a list of possible analyses engines. When input is received from the user choosing the residence of the container **11300**, the program restores the search interface screen.

Referring now to FIG. **6**, a flow chart of the method for searching for containers **100**.

When input is received from the user selecting "search interface" from the main title bar, the search interface screen appears. The user is given the choice of containerizing selected content or requesting that container levels be displayed **30100**. From a drop down menu another menu appears allowing input to be received from the user selecting the container level **30200**. Input is received from the user selecting the container level (from the smallest component to the whole system) **30300.**

Input is received **30310** from the user selecting the phrases, containers or components, which then are re-submitted to the same process, until the input is received from the user selecting a specific site or container.

The search phrase, whether containerized or not, is submitted simultaneously to the search engine **30400** and the analysis engine **30500**.

The screen then reports in a selection menu, the number of applicable sites found by the search engine **30410**, the number of historically proven applicable sites found by the analysis engine **30410**, the number of historically proven applicable containers at the selected container level or any container level found by the analysis engine **30410**, and the number of historically proven new search phrases or digital segments found by the analysis engine **30320**. . Input is received from the user selecting one of the named sets above **30330**. If input is received from the user choosing the search engine, the search interface lists the applicable site titles with a brief description **30410**. If input is received from the user choosing the site list of the analysis, the search interface lists the applicable site titles with a brief description **30410**. . If input is received from the user choosing the container list of the analysis engine, the search interface lists the applicable container titles with a brief description **30410**. . If input is received from the user selecting a container **30420**, the system offers the means to view titles and descriptions of sub-containers at any chosen class level. . If input is received from the user choosing the phrase list of the analysis engine, the search interface lists the applicable phrases or digital segments with a brief description **30320**. The search and search result cycle repeats until input is received from the user choosing to go to an individual container or site.

Input is received from the user entering text or any digital string describing his search objectives into a text or search box. When input is received from the user submitting the search string, the system provides the option of containerizing the search through the container editor **110.** Once the search container **101** is created, the system restores the search interface **300** screen the user.

Input is received from the user selecting "search", "supported search" or "both" from another drop-down menu and from submitting the search. When input is received from the user selecting "search" 30310**,** the search phrase is submitted to the search engine **30400**, which searches both content and the appropriate container registers, as pre-indexed in the search engine, and returns a list of appropriate locations, components or containers. When input is received from the selecting "supported search", the search phrase is submitted to the analysis engine search support, which returns a list, in a drop-down menu, of search phrases or individual containers, for any and all container levels, used by other users or created by the system and known to be historically successful for the described effort and the described searching user, as per the results of the analysis search engine. Input is received from the user selecting a new search phrase or specific container from the drop down menu **30330**. When input is received from the user choosing a new search phrase, that phrase is also submitted to the analysis engine **30500** which returns a list of pre-compiled historically proven sites, components or containers associated with that search phrase **30320**. Input is received from the user choosing a selection **30420** and the system calls up that specific site, container or component. If input is received from the user selecting a specific site, container or component at any time during the search process, that element is called up by the system **30440**.

Input is received from the user choosing to containerize a search or select a container level in which to search **30100**. When input is received from the user choosing to containerize the search, the software moves to the container editor as described in Fig. **5**, and then returns the user to the search interface screen. Input is received from the user selecting to search a specific container level or the whole network. The system shows the available levels **30200**. Input is received from the user selecting a container level **30300**, and entering the text or digital component comprising the search string **30310**. The system searches the containers **30400** while simultaneously submitting the search string to the analysis engine **30500**. While the system is accessing containers, sites or templates **30700**, the analysis engine **30500** inquires of the appropriate database **30600** to access historically successful containers, sites or search templates corresponding to the search request **30700**, which is then shown on another portion or option of the search interface, either as available containers or sites **30410** or as search template options **30320.** On one portion or option of the search interface screen the corresponding containers or sites are listed and/or previewed for selection **30410**. Input is received from the user selecting the container to access **30420**. The system accesses that container **30430** and shows it on the screen **30440** for user review. Input is received from the user selecting an operation, i.e., preview, read, purchase, move, copy, lease, in any composed schedule with operations assigned specific values **30460**, and the system obtains the specified result **30470**. The selection of the operation including any interaction with any uniquely defined container **100** is recorded **30800** by the container gateway (Fig. 2 A, **200**), stored in the gateway storage **205** and made available to the analysis engine (Fig. **9**) by the data collection and reporting means (Fig. **8**). Reporting and collection occurs on a regular basis according to user determined times or rules. The analysis engine compiles and analyzes selections according to various rules-based systems applicable to the particular container area of residence in cyberspace.

. Input is received from the user selecting the container or site **30410**, proceeding as described above, or selecting a search template **30330**, and editing it to re-enter the search

30310. All operations on Fig. **6** utilize the communication device **26** whenever necessary or expeditious.

Referring now to FIG. **7**, a flow chart of the search process is shown. Steps in FIG. **7** repeated from FIG. **6** are given the same reference number as in FIG. **6** for convenience and ease of understanding. Fig**.** **7** commences with "SEARCH TRANSITS GATEWAY **32100**", continuing from Fig. **6**, "SYSTEM SEARCHES CONTAINERS **30400**". The submitted search **32100** transits the gateway **200**. The gateway **200** interacts with the container registers **32200**. The gateways **200** store the information downloaded from the registers **32300**, and the container registers are altered **32500**. The container registers **120** then interact with the registers **120** of the encapsulated search, which registers, and the values set within, have been constructed and appended to the search through the search interface **32600.** Values are exchanged and compared and operations performed under the rules governing both interacting containers **100**, and the rules governing the search container **100** and any gateway **200.** The search engine **320,** operating under the principles and means of search engines presently existing as described elsewhere, **then** provides to the search interface **32600** a list of containers **100** meeting the requirements of the search and its appended registers, as well as additional search options **32900**. The gateway **200** reports and makes available for collection to the analysis engine **400** the information obtained from the interaction **32400**. On a periodic basis defined by the user or a rules-based system, the analysis engine **400** (Fig. **9**) stores in databases **900**, analyzes and instructs the execution engine **500**, and the execution engine **500** executes changes in the system components as defined below. (Fig. **10**). All operations on Fig. **7** utilize the communication device **26** whenever necessary or expeditious.

On the remaining figures, shapes referring to other figures, to operations external to the scope of the present figures, or to the subject of the present drawing, are indicated with dashed lines, and are shown only to place the described operations in the context of continuous and continual operations external to the drawing.

Referring now to FIG. **8**, a flow chart of the preferred process for collecting and reporting information on containers is shown. The data reporting **600** and data collection **700** means utilizes subroutines within the analysis engines **400** and gateways **200** to submit and collect register information and sub level analysis to other analysis engines **400** or other gateways **200** of a higher (larger) logical set in a set pattern and frequency defined by the administrator.

Input is received from the user selecting "data reporting" **70100** from the "edit gateway" drop-down menu. Container levels are displayed **70200**. Input is received from the user selecting container level **70300.** A menu of all possible gateways **70320** and analysis engines **70330** residing on gateways on the above defined container class appears, depicted graphically as a tree of analysis engines and gateways at that container level. Input is received from the user selecting "source" from "source or destination." Input is received from the user **70400** selecting a container, containers, or class of container by clicking on the graphically depicted container(s) or container level on a display device. Input is received from the user **70410** selecting "destination" from "source or destination" Input is received from the user **70500** selecting an analysis engine, analysis engines, or class of analysis engine by clicking on the graphically depicted analysis engine(s) or analysis engine level on a display device. A time scheduler is displayed. Input is received from the user **70510** selecting the reporting frequency for the selected gateways to report data to the selected engines. The data from the gateways is thenceforth continuously moved or copied to the analysis engines by the system **10** utilizing the execution engine **500** according to the defined schedule, rules and pattern **70420, 70520.**

Input is received from the user selecting "choose container level" **70300** from the gateway editor drop-down menu. A menu **70320** appears listing the classes of containers on the system within the defined subset and superset scheme of multiple hierarchically nested containers, i.e.; element, document, file, database, warehouse, domain, appears. Input is received from the user selecting the class of containers. A graphic representation of that container level throughout the system appears. Input **70300** is received from the user selecting individual containers or all the containers in that class.

From the gateway editor drop-down menu input **70100** is received from the user selecting "data collecting" A menu of all possible gateways and analysis engines residing on gateways on the above defined container class appears, depicted graphically as a tree of analysis engines, and gateways at that container level. Input **70510** is received from the user selecting "source" from "source or destination." Input is received from the user selecting a container, containers, or class of container by clicking on the graphically depicted container(s) or container level. Input **70510** is received from the user selecting "destination" from "source or destination." Input **70510** is received from the user selecting an analysis engine, analysis engines, or class of analysis engine by clicking on the graphically depicted analysis engine(s) or analysis engine level. A time scheduler appears. Input **70510** is received from the user selecting the collecting frequency for the selected engines to collect data from the selected gateways. The data from the gateways is thenceforth continuously moved or copied to the analysis engines by the system **10** utilizing the execution engine **500** according to the defined schedule, rules and pattern.

The data collection **700** means, utilizing the communication device **26** and an execution engine **500**, comprises one or more subroutines or agents programmed to travel through the network collecting the accumulated data and analyses from selected analysis engines, gateways or selected subset level of analysis engines or gateways (as above) in a pattern and frequency defined by the gateway administrator at a given container level. Input **70510** is received from the user or administrator, defining the collection and reporting of data, thus controlling permission within his gateway, and being subject to permission levels defined by others beyond his gateway.

Input is received from the user or gateway administrator selecting collection or reporting **70100** and the system shows the container levels available **70200**. Input is received from the user selecting a container level **70300**. Input is received from the user selecting "gateway" **70400** or "engine" **70500**. The system shows gateways **70320** or engines **70330** associated with that level. Input is received from the user editing the reporting parameters associated with a gateway or a class of gateways **70410** or an engine or class of engines **70510**. Input is received from the user selecting the collecting frequency for the chosen engines. When input is received from the user choosing to user save the definition, the screen returns to the main container screen, step **70100** to make another selection available. Input is received from the user choosing to repeat the cycle, choosing "destination" to describe the destination analysis engines and the data collecting frequency from those destination analysis engines. The data collection means **700** collects the accumulated gateway information in a pattern and frequency defined by the gateway administrator or user at a given container level.

The system utilizing the execution engine (see Fig. **10**) distributes the new parameters to the gateways **70420** or engines **70520** by the communication device **26**. Using the new parameters the gateways report to the analysis engines **70430** after, in some cases, conducting sub-analysis **70440,** or using sub-analysis **70440** to submit directly to specified gateways under certain conditions and parameters, and the analysis engines collect from the gateways **70530.**

The analysis engine uploads, downloads and utilizes information to databases **900** to conducts its analysis.

The invention includes an analysis engine **400**. Through the data reporting **600** means and data collection **700** the analysis engine **400** receives data and sub-analysis from the search interface and the gateways. Data includes, for each gateway **200**, the frequency and grade of access, the description of the user accessing, the identity of the container **100** accessing, the register parameters, and the historically accumulated register data.

Referring now to FIG. **9**, a flow chart of the operation of the analysis engine **400** is shown. Analysis engines **400** may reside at any gateway or anywhere in the system 10. The analysis engine **400**, operating under its own programmed sequence, utilizing the communication device **26**, works, by means of programmed rules of logical, mathematical, statistical or other analysis upon gateway and register information, in continuous interaction with the search process **410** and the data collection and reporting process **420** to analyze, determine and compile instructions **40100** on container construction **40110** to containerize in an automated process **40115**, on container contents **40120** to move, copy or delete containers **40125**, on storage schemes **40130** to move or copy containers to new storage **40135**, on access routes **40140** to alter gateway pointers to sought information **40145**, on search templates **40150** to add, delete or change search phrases and the referenced objects indicated by those search phrases **40155** and on gateway instructions **40160** to alter gateway registers and pointers **40165**.

Thus, analyses might include, but are not limited to, the physical locus of the users accessing, the demographic classification of the users accessing, the access frequency for a given container, the range or curve of time relevance affecting a container, the range or region of space relevance affecting a container **100**, the number or number of a specific type of container **100** transiting a gateway **200**, the hierarchically graded usage of containers **100** or container contents **01** compared with the demographic of those users accessing the container, the hierarchically graded usage of containers **100** or container contents **01** compared with search phrases entered into the search interface **300**, the hierarchically graded usage of containers **100** or container contents **01** compared with search phrases entered into the search interface 300 compared with the demographic of the users accessing, the number of pertinent containers nested within a given container **100**. Once an analysis is accomplished, the result is compared to pre-programmed rules triggering instruction sets (such as moving a container to nest within another container).

Instructions are then sent to the execution engine **40200**, which utilizes the communication device **26** to execute the instructions derived from the analyses. These containerized instructions transit the gateways **40300** and are utilized in the gateway process (Fig. **12**)

Referring now to FIG. **10**, a flow chart of the operation of the execution engine is shown. The execution engine **400**, operating under its own programmed sequence in response to the instructions from the analysis engine **50100**, utilizing the communication device **26**, works in continuous process as its containerized execution instructions transit the gateways **50200** to create containers **50210** in an automated containerization process **50215**, alter container contents **50230** by moving or copying containers to new containers **50235**, to alter storage **50240** by moving or copying containers to new storage **50245**, to alter access routes **50250** by altering gateway pointers **50255**, to alter search templates **50260** by adding, changing and deleting search phrases and the referenced objects indicated by those search phrases **50265**, to alter gateway instructions **50270** by altering gateway registers and pointers **50275**. The execution works in a continuous loop with the gateway process **50300**, the data collection and reporting process **50400** and the analysis engine process **50300**.

The invention includes gateways **200**. Gateways may be placed and reside anywhere on the network where containers transit. Gateways also reside on any or all containers. The gateway reads and stores the chosen register information from transient containers entering or exiting its logical boundaries. The resident analysis search engine, if any, performs the specified level of analysis. Data and analysis is both held for the collection means according to the pattern and timing specified in the data reporting **600** editor and submitted according to the pattern and timing specified in the data collection means editor **700.**

The gateways are network-wide, hierarchical, and nestable, and reside with a container encompassing any component, digital code, file, search string, set, database, network, event or process and maintaining a unique lifelong network wide identity and unique in all the universe historical identity, or may be strategically placed at such container transit points to gather and store register information attached to any such container, according to system-defined, system-generated, or user determined rules residing in its registers defining the behavior of those containers and components as they exit and enter one another, or interact with one another or any system process or system component within the logical domain of that container, or after exiting and entering that container, or defining how they interact with that unique gateway.

Gateway's registers comprise both system-defined and user-defined registers, alterable by author, duration, location, network-wide history, individual container history and/or interaction with other containers, gateways, networks or media, and evolve according to that gateway's history on a computer network, or according to the network history of events and processes, or according to that information component's interaction with other information containers, components, system components, network events or processes.

Referring now to FIG. **11,** a flow chart of the gateway editor is shown. From the main title bar input is received from the user selecting "containerize" or "gateway level" **20100.** When input is received from the user selecting "containerize" the system enters the container editor process **110.** When input is received from the user selecting "gateway," the system shows the gateway levels available **20200.** A menu of all possible gateways within the subset and superset scheme of defined multiple hierarchically nested gateways appears. Input is received from the user selecting the gateway level **20300.** The system searches the gateways **20500** to locate the available gateway templates **20700** and the available gateways **20600.** Input is received from the user selecting the gateway **20610** or gateway level template **20720.** The system goes to the gateway **20620** or to the template **20720.** A graphic representation of the chosen gateway **20630** or template **20730** appears. Input is received from the user to edit **20640** or create a gateway **20740.** Once completed, input may be received from the user selecting "analysis level" from the gateway **200** drop-down menu, to select the level of analysis in a multi-level analysis sequence to be accomplished at the local level by a gateway-resident analysis engine. The user accesses the container editor to containerize (Fig. **5**). Input is received from the user selecting the registers by clicking on the graphically depicted register, or from a drop down menu.). Input is received from the user setting the registers as described elsewhere in ("container registers"). Input is received from the user selecting or defining the rules governing the interaction of that gateway with transient containers. Input is received from the user selecting or defining the rules governing the interaction of containers existing within the logical domain of the container 100 to which that gateway is attached. The user publishes the gateway (Fig. **5**). Input is received from the user selecting "residence" from the main menu bar.

). Input is received from the user choosing to leave the gateway where it was created, move it to container on another drive, directory, computer, or network. If the user chooses "move," a browse function allows the user to name the new location or browse a list of possible locations. Once input is received from the user choosing the residence of the gateway, the program restores the search interface screen.

The invention includes a data reporting means editor **610**, and a data collection means editor **710,** Fig. 2 A, as a menu option under the gateway editor **210**.

The present invention also includes a gateway process.

Referring now to FIG. **12**, a flow chart of the gateway process is shown. A system operation, search process or element container or process container is shown in transit **21100** passing through a gateway **21200**. The container, operation or process interacts with the gateway **21300**, uploading, downloading and exchanging information with the container, operation or process. The gateway stores container information **21400** and the container registers are altered **21500**. The container registers also interact with the search interface **21600**. The gateways report the register information or make it available for collection by the data reporting and collection means (Fig. **8**) operating on the communication device **26** to provide the information to the analysis engine **21800**, which stores **90100**, analyzes and instructs the execution engine **21900**, which processes and instructions are also stored **90100** by the execution engine upon receipt.

All operations in Fig. **12** utilize the communication device **26** whenever necessary or expeditious.

Referring now to FIG. **13 A,** a drawing of nested containers **100** prior to the container modification process on a network **201** is shown. (Note: The same container numbering scheme is used in Fig. 13 A, 13 B, 13 C, 13 D and in 2 B.) Information containers **505** and **909**, residing within container **908**, operating under the rules governing container interaction within that container **908** downloaded to container **505** and **909** from gateway **9081** upon their entrance to container **908,** which rules had been downloaded from execution engine **500** acting under the direction of analysis engine **400**, and under the rules programmed into their own registers **404120**, **909120**, compare the specified (by those rules) set of registers **404120**, **909120**, i.e., time and space, and determine a container **404** encapsulated within **505** would be more appropriately encapsulated within container **909**.

Referring now to FIG. **13 B** a drawing of nested containers during a container modification process on a network **201** is shown. Container **404** is moved to reside with container **909**. As the container **404** exits container **505**, the gateway of container **505**, being gateway **5051**, operating under the rules governing container interaction with a gateway 5051 upon egress or egress as programmed in the gateway editor **210** and modified by the execution engine **500** executing the instructions of the analysis engine **400**, or any greater logical analysis engine **408** providing execution instructions to an execution engine **508** operating in a larger encompassing container **108** entering through that container's gateway **208** or an independent gateway **707**, or sub-analysis engine operating at any gateway level, records the register information of container **404.** The gateway **5051** reports the transaction to the gateway **9081** of container **908**, being the next higher logical container. Gateway **9081** holds in gateway storage **205** the information until collected by one or more data collection processes **700**, or reported to one or more data reporting processes **600**, serving one or more analysis engines **400** residing independently on the system **10** or an analysis engine at higher logical container **303**. The analysis engine **400,** comparing reports of user hierarchically graded usage under the operations of the search engine **320** and the search interface **300**, on information container **808** after receiving reports from the data reporting means of container **404** being moved to container **909** determines, i.e., that the number of time and space relevant containers residing within container **909** is sufficient to warrant an action, and directs the execution engine **500** to copy container **909**, nested within container **908**, to a third information container **808**. As the copy instruction from execution engine **500** transits the gateway of container **908**, the gateway **9081** records the instruction. The copy instruction interacts with the registers **909120** of container **909** regarding the rules governing its copying to another location. Once approved by the governing rules of registers **909120** appended to container **909**, container **909** is duplicated. As the duplicate container **909** exits the container **908**, the gateway records the register information **909120** of container **909**, and the registers **909120** of container **909** are altered by special instructions from gateway **9081** under the rules residing in gateway **9081** regarding ingress and egress and the rules residing in the registers **909120** of container **909** regarding alteration by gateways upon ingress and egress. Passing through independent gateway **707**, the register information **909120** is recorded, and awaits data collection or reporting 700, **600.** As container **909** enters container **808**, the gateway records the register information **909120** of container **909**, the registers **909120** of **909** are altered by special instructions from gateway **8081**, operating under the rules as described in the paragraph above, and container **909** takes up residence within container **808.**

Referring now to FIG. **13 C,** a drawing of nested containers after the container modification process on a network **201** process is shown. Container **909**, now also logically residing within container **808,** commences to interact with other containers **606** in **808** under the rules governing container interaction within container **808** as received from gateway **8081** upon transiting that gateway, and under the rules of registers **606120, 909120 of** the interacting containers **606, 909**, operating under the rules as described in the paragraph above. Through data collection and reporting **700, 600**, analysis engine is appraised of container's **909** new duplicate residence. I.e., operating under the registers of space relevance, a body of law pertaining to Boston Municipal tax law may be housed in a container holding Massachusetts tax law, but it would be more appropriately located in a container holding Boston tax law, with only a pointer to that location residing in the Massachusetts tax law container. In this example, such an analysis could be accomplished by comparison of zip code information in the space registers, or logical rules-based analysis, with "state" being a larger set than "city". Or, i.e., operating under the registers of time relevance, the curve of time relevance for a concert might follow an ascending curve for the months prior, hit a brief plateau, and then reach a precipitous decline, at which time certain pertinent information only might be moved to an archival container of city events or rock concerts of that year. In this example, once the curve is mapped into a register, that map would cause an increasing frequency of pointers to that container in other containers or gateways, or inclusion of that container in other containers, as the analysis engine compares that curve with increasing user inquiry.

Referring now to Fig. **13 D**, a flowchart of the reconstruction process is shown.

Information containers **505** and **909**, residing within container **908**, operating under the rules governing container interaction within that container **908** downloaded **888103** to container **505** and **909** from gateway **9081** upon their entrance to container **908,** which rules had been downloaded **888102** from execution engine **500** acting under the direction **888101** of analysis engine **400**, and under the rules programmed into their own registers **404120, 909120**, compare **888104** the specified (by those rules) set of registers **404120, 909120**, i.e., time and space, and determine **888105** a container **404** encapsulated within **505** would be more appropriately encapsulated within container **909**.

Container **404** is moved **888106** to reside with container **909**. As the container **404** exits container **505**, the gateway of container **505**, being gateway **5051**, operating under the rules governing container interaction with a gateway **5051** upon egress or egress as programmed in the gateway editor **210** and modified **888108** by the execution engine **500** executing the instructions of the analysis engine **400**, or any greater logical analysis engine **408** providing execution instructions **888107** to an execution engine **508** operating in a larger encompassing container **108** entering through that container's gateway **208** or an independent gateway **707**, or sub-analysis engine operating at any gateway level, records **888109** the register information of container **404**, and alters the register information of container **404.** The gateway **5051** reports **888110** the transaction to the gateway **9081** of container **908**, being the next higher logical container. Gateway **9081** holds **888111** in gateway storage **205** the information until collected by one or more data collection processes **700**, or reported to one or more data reporting processes **600**, serving **888112** one or more analysis engines **400** residing independently on the system **10** or an analysis engine at higher logical container **303**. The analysis engine **400,** comparing **888114** reports of user hierarchically graded usage on information container **808** under the operations of the search engine **320** and the search interface **300**, after receiving **888113** reports from the data reporting means of container **404** being moved to container **909**, determines **888115**, i.e., that the number of time and space relevant containers residing within container **909** is sufficient to warrant an action, and directs **888115** the execution engine **500** to copy container **909**, nested within container **908**, to a third information container **808**. As the copy instruction from execution engine **500** transits the gateway of container **908**, the gateway **9081** records **888116** the instruction. The copy instruction interacts **888117** with the registers **909120** of container **909** regarding the rules governing its copying to another location. Once approved **888118** by the governing rules of registers **909120** appended to container **909**, container **909** is duplicated **888118**. As the duplicate container **909** exits the container **908**, the gateway records **888119** the register information **909120** of container **909**, and the registers **909120** of container **909** are altered **888120** by special instructions from gateway **9081** under the rules residing in gateway **9081** regarding ingress and egress and the rules residing in the registers **909120** of container **909** regarding alteration by gateways upon ingress and egress. Passing through independent gateway **707**, the register information **909120** is recorded **888121**, and awaits **888122** data collection or reporting **700, 600.** As container **909** enters container **808**, the gateway records **888123** the register information **909120** of container **909**, the registers **909120** of **909** are altered **888124** by special instructions from gateway **8081**, operating under the rules as described in the paragraph above, and container **909** takes up residence **888125** within container **808.**

Container **909**, now also logically residing (in addition to its original container residence) within container **808,** commences to interact **888126** with other containers **606** in **808** under the rules governing container interaction within container **808** as received from gateway **8081** upon transiting that gateway, and under the rules of registers **606120, 909120 of** the interacting containers **606, 909**, operating under the rules as described in the paragraph above. Through data collection and reporting **700**, **600**, analysis engine is appraised **888127** of container's **909** new duplicate residence.

Referring now to Fig. **14**, the screen interface of the container editor is shown. This interface is a process wherein input is received by the user using the main menu **78** or drop down menu **1419**, or using an input device to "drag and drop" or click, causing the system **10** to acquire **1409**, edit **1410** or create **1411** a file **1407**, container **1408** or digital content **01,** to search for **1412**, acquire **1413**, edit **1414** or create **1415**, print **1416**, or containerize **1417** a container **100**, to select **1402**, (or by clicking on register), search **1403**, acquire **1404**, edit **1405**, or create a register **1406** to append or detach registers **120** to those containers, to set register values in those registers **120**, to utilize the register editor **125** through **1405** to create new registers, or to **1418** add, detach, acquire a gateway **200** to append or detach to those containers, and utilize the gateway editor **210** through **1418**. (See detailed description referring to Fig. 5)

Referring now to Fig. **15**, the screen interface of the gateway editor is shown. This interface is a process wherein input is received by the user using the main menu **1501** or drop down menu **1513**, or using an input device to "drag and drop" or click, causing the system **10** to search for **1507**, acquire **1508**, edit **1509** create **1510**, print **1511** or containerize **1512** gateways, and causing the system **10** to establish rules by which an individual gateway governs the transiting **1502**, entering **1503**, exiting **1504** of containers and the interaction of containers within its domain **1505**, and external of its domain.**1506**. (See detailed description referring to Fig. **11**).

Referring now to Fig. **16**, the screen interface of the search interface. This interface is a process wherein input is received by the user using the main menu **1625** or drop down menu

1624, or using an input device to "drag and drop" or click, or by entering text, causing the system **10** to select **1615**, search for **1616**, acquire **1617**, edit **1618** create **1619**, print **1620**, containerize **1621** (by accessing the container editor **110)** or insert **1622** digital search strings into the search box **1623** in order to submit that string to the search engine **320**, or causing the system **10** to select **1602**, search for **1603**, acquire **1604**, edit **1605**, create **1612**, containerize **1613** (by accessing the container editor **110**), or insert **1614** search keys (templates that comprise search scope in geographic range, container level, and specific key words or digital strings), or containerized searches (containers **110**), into the search box **1623** in order to submit that string to the search engine **320** , or causing the system **10** to set a search range by geographic range **1607**, container level **1608**, or acquire **1609**, edit **1610** or create **1611** a scope template. (templates that comprise search scope in geographic range and, container level.) (See detailed description referring to Fig. 6).

Referring now to Fig. **17**, a drawing showing, on an input device or computer screen **24**, in any generic (dashed lines) software application program, a drop-down menu link **1403** on a drop down menu **1402** dropping down from a main menu **1401**, and a free-floating button link **1404**, is shown. When input is received at **1402** or **1403**, the system **10** makes available to the user the containerization process or container editor **110.** When input is received at drop-down menu link **1405** or a button link **1406**, the system **10** makes available to the user the means to enter and interact with this system **10** or this network **201** in any of their aspects. The interfaces **1403, 1404** show a process wherein input is received causing the system **10** to encapsulate content or access the container editor **110**. The link also allows the user to encapsulate the page or file on which he is currently working, without selecting content, and if so desired, without accessing the container editor. The interfaces **1405, 1406** show a process wherein input is received causing the system **10** to access or interact with the system **10** or the network **201**.

The present invention also includes a search engine **320**. Once the key word(s), phrase or digital segment is entered into the search interface **300,** or an offered selection chosen on the menu, it is utilized by the search engine **320** to locate the desired site or data.

The search engine employed may be any industry standard search engine such as Verity "Topic", or Personal Library Software, as used in Dow Jones News Retrieval, or Internet search engines such as Webcrawler, Yahoo, Excite, Infoseek, Alexa or any Internet search engine, or any new engines to be developed capable of searching for and locating digital segments, whether text, audio, video or graphic.

The present invention also includes an analysis engine **400.** Utilizing rules-based analysis, the analysis engine determines the class of storage medium upon which containers reside, the subsets and supersets by which and in which containers encompass and reside within one another, the routes of access to those containers, the historically successful search parameters by which those containers are accessed based upon the identity of the user accessing the containers, and the grade of access chosen by the user in accessing that container **100**.

Utilizing a pre-programmed sequence of compilation, and inductive, deductive and derivative analysis, the analysis engine manufactures instructions based upon the analysis of the information submitted by the gateways and the search interface, and submits those instructions to the appropriate execution engine **500** in order to create new information containers, content assemblages, storage schemes, access routes, search templates, and gateway instructions, and others, and to provide informed search options through the search interface to the inquiring user.

The present invention also includes an engine editor **510**, that provides a system administrator with a means of editing the operating principles of that search engine, and search template loading in the search interface **300**, a reporting and collection means editor **610**, **710**, governing data reporting **600** and data collection **700** at the gateways **200** as defined by the gateway editor **210** and the register editor **125**, a container editor **110** for creating and modifying containers and appending registers to containers, a register editor **125** for creating and modifying container registers and establishing and adjusting the values therein, container gateways **200** with their own storage **205**, information containers **100** for holding information and container registers for holding information about specific containers and their history on the network.

The present invention also includes an execution engine **300**. Based upon instructions received from the analysis engine **400** utilizing the communication device **26**, the execution engine **500** provides search phrases to the search interface **300** based upon initially received inquiries, relocates containers including their programs, data and registers to other directories, drives, computers, networks on other classes of storage mediums, i.e., tape drive, optical drive, CD-ROM, deletes, copies, moves containers to nest within or encompass other containers on other directories, drives, computers, networks to nest within other containers, alters the class of storage medium upon which containers reside, the subsets and supersets by which and in which containers encompass and reside within one another, the routes of access to those containers, and the historically successful search parameters by which those containers are accessed based upon the identity of the user accessing the container and the grade of access chosen by the user in accessing that container.

The execution engine **400** fulfills the instructions of the analysis search engine **500**, to create new information containers, content sub and superset assemblages, storage schemes, access routes, search templates, gateway **200** instructions and other system functions. The execution engine includes an editor **510** that provides a system manager with a means of editing the operating principles of that search engine, governing data reporting, data collection **700**, search template loading, gateway instructions, and other functions.

The present invention also includes flat or relational databases **900**, used where, and as required.

The present invention also includes a communication device **26** supporting all operations on a network wide basis.

The present invention also includes a search engine **300** to locate the desired site or data. The present invention also includes databases **900**, flat or relational, to serve the other components of the system as needed and where needed.

The present invention also includes editors, by which the user may alter the governing aspects of the system. Editors include, but are not limited to, a container editor **110**, a register editor **125**, a gateway editor **210**, an engine editor **510**, a reporting means editor **610**, a search interface **300**, and a collection means editor **710**.

The present invention also includes specific screen interfaces for the editors, as described in Fig. **14**, Fig. **15**. and Fig. **16**.

The present invention also includes a means for this system **10** and network **201 or container editor 110** to be accessed from a menu or button selection within any program, as described in Fig. **17**.

While the present invention has been described with reference to certain preferred embodiments, those skilled in the art will recognize that various modifications may be provided. For example, both analysis engine and execution engine may be duplicated or modified for distribution at various locations and hierarchical positions in the gateway and container system throughout the network and designed to work in concert. Also, the physical computing infrastructure may be mainframe, mini, client server or other with various network and distributed computing designs, including digitally supported or based physical or public media, and the components of the system 10, as described in Fig. 1 may be physically distributed through space. Even the contents of a single container may be logically referenced but be physically distributed through the network and reside at multiple storage locations. The whole system may be hierarchically nested within other systems to the nth degree. Whole systems may also be encapsulated within containers. A single container may also encompass a single physical media, such as a CD-ROM disk, programmed with the container, gateway and register design. Gateways may be strategically placed on containers at ingress and/or egress points or may be placed strategically throughout the system for optimal collection and reporting output and gateway system control. Also, the loop of gateway data collection and reporting, analysis engine analysis, instruction, and gateway modification, and execution engine operations may be infinitely nested, from the smallest container of two sub-containers to whole networks holding millions of containers and thousands of levels, with analysis itself nested within the multiple levels. Gateways may be established at both logical and physical junctures such as a satellite uplink point. Also, the provision to establish a unique network identity might be designed to include as of yet unknown computer networks as they arise. The analysis and execution engines may operate on a rules-based, fuzzy logic, artificial intelligence, neural net, or other system not yet devised. Other variations upon and modifications to the preferred embodiments are provided for by the present invention, which is limited only by the following claims. Also, the classification scheme of nested containers, while designated by the container creators, may transform, be utilized otherwise, or be wholly discarded according to usage. Also, hardware configurations, such as the use of RAM or hard drives for storage or lasers for communication may assume myriad forms without altering the essential operation of this invention.

WHAT IS CLAIMED IS:

1. An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element;

a plurality of registers, the plurality of register forming part of the container, a first register of the plurality of registers for storing a unique container identification value, a second register of the plurality of registers that stores information and evolves according to the relationship, use and interaction of the container with other containers, processes and systems; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems and process.

2. The apparatus of claim 1, wherein the information element is one from the group of text, graphic images, video, audio, a digital pattern, a process, a nested container, bit, natural number and a system.

3. The system of claim 1, wherein the plurality of registers include at least one container history register for storing information regarding past interaction of the container with other container, system or processes, the container history register being modified.

4. The system of claim 1, wherein the plurality of registers include at least one system history register for storing information regarding past interaction of the container with different operating system and network processes.

5. The system of claim 1, wherein the plurality of registers include at least one predefined register, the predefined register being a register associated with an editor for user selection, the predefined register appendable to any container.

6. The system of claim 1, wherein the plurality of registers include a user-created register, the user-created register generated by the user, one or more of which is appendable to any container.

7. The system of claim 1, wherein the plurality of registers include a system-defined register, the system-defined register set, controlled and used by the system, one or more of which is appendable to any container.

8. The system of claim 1, wherein the plurality of registers include at least one register for controlling the relationship of the container with other containers, systems and processes using time as a parameter.

9. The system of claim 8, wherein the plurality of registers include:

an active time register for identifying times at which the container will act upon other containers, processes, systems or gateways;

an passive time register for identifying times at which the container can be acted upon other containers, processes systems, or gateways; and

a neutral time register for identifying times at which the container may interact with other containers.

10. The system of claim 1, wherein the plurality of registers include at least one acquire register for controlling whether the container adds a register or a container from other containers when interacting with them.

11. The system of claim 1, wherein the plurality of registers include at least one register for controlling the relationship of the container with other containers using space as a parameter.

12. The system of claim 11, wherein space refers to the geographic location of a the container.

13. The system of claim 11, wherein space refers to the logical address space of a network in which a container resides.

14. The system of claim 11, wherein the plurality of registers include:

an active space register for identifying space in which the container will act upon other containers, processes, systems or gateways;

an passive space register for identifying from which the container can be acted upon other containers, processes systems, or gateways; and

a neutral time register for identifying space in which the container may interact with other containers.

15. The system of claim 1, wherein the gateway includes means for acting upon another container, the means for acting upon another container using the plurality of register to determine whether and how the container acts upon other containers.

16. The system of claim 1, wherein the gateway includes means for allowing interaction, the means for allowing interaction using the plurality of registers to determine whether and how another container can act upon the container.

17. The system of claim 1, wherein the gateway includes means for gathering information, the means for gathering information recording register information from other containers, systems and processes that interact with the container.

18. The system of claim 1, wherein the gateway includes means for reporting information, the means for reporting information providing register information to other containers, systems and processes that interact with the container.

19. The system of claim 1, wherein the gateway includes an expert system including rules defining the interaction of the container with other containers, systems and processes.

20. A method for creating an interactive information container, the method including the steps of:

forming a container;

selecting an interactive register for the container;

identifying an item for inclusion in a container; and

creating a container element that includes the identified item.

21. The method of claim 20, wherein the step of forming a container further comprising the steps of:

displaying a plurality of container levels;

receiving input from a user selecting one of the displayed container level; and

displaying a container template corresponding to the container level input.

22. The method of claim 20, wherein the step of selecting an interactive register further comprising the steps of:

displaying a list of available registers;

receiving input selecting an available register from the list of available registers;

receiving input values for the selected available register; and

appending the register to the container.

23. The method of claim 20, wherein the step of creating a container element that includes the identified item further comprising the steps of:

providing a data structure as part of the container element;

storing the identified element in the data structure; and

associating the container element with the container.

24. The method of claim 20, wherein the step of forming a container includes the step of providing for the container a gateway that uses the interactive register to control the interaction of the container with other containers, processes, and systems.

25. The method of claim 24, wherein the step of providing a gateway further comprising the steps of:

determining a current gateway for a system upon which the container is being created;

replicating the current gateway to create a new gateway; and

associating the new gateway with the container.

26. The method of claim 24, wherein the step of providing a gateway further comprising the steps of:

determining a register for a system upon which the container is being created;

replicating the determined register to create a new register; and

associating the new register with the container.

27. The method of claim 20, wherein the step of selecting an interactive register further comprising the steps of:

retrieving a list of available registers;

selecting an available register from the list of available registers by the system;

receiving input values for the selected available register from the system; and

appending the register to the container.

28. The method of claim 20, wherein the step of creating a container element that includes the identified item is performed by a system interacting with the container, and further comprising the steps of:

providing a data structure as part of the container element;

storing the identified element in the data structure; and

associating the container element with the container.

29. A method for interacting between a first interactive information container and a second interactive information container, the method including the steps of:

determining identification information for the first container using a first gateway;

determining identification information for the second container using a second gateway;

determining whether the first container can act upon the second container using the first gateway and a register of the first container;

determining whether the second container can be acted upon by the first container using the second gateway and a register of the second container; and

performing the interaction between the first and second containers prescribed by the first gateway and the register of the first container if both the first container can act upon the second container and the second container can be acted upon by the first container.

30. The method for interacting of claim 29, wherein the steps of determining identification information are performed by reading respective identification registers of the first and second containers.

31. The method for interacting of claim 29, further comprising the step of altering a register of the first container and a register of the second container to reflect the interaction between the first container and the second container.

32. The method for interacting of claim 29, further comprising the step of adding registers to the first container based on the registers in the second container and the second gateway.

33. The method for interacting of claim 29, wherein the step of performing also uses the second gateway and the register of the second container to determine the prescribe action to be taken.

34. The method for interacting of claim 29, further comprising the steps of:

determining whether the first container should add an identified register of the second container as a new register of the first container using an acquire register and the first gateway of the first container; and

adding the new register to the first container if it is determined that the new register should be added to the first container.

35. The method for interacting of claim 29, further comprising the step of modifying the first gateway of the first container based on the interaction between the first container and the second container.

36. The method for interacting of claim 35, wherein the step of modifying includes modifying rules of an expert system that forms the first gateway of the first container.

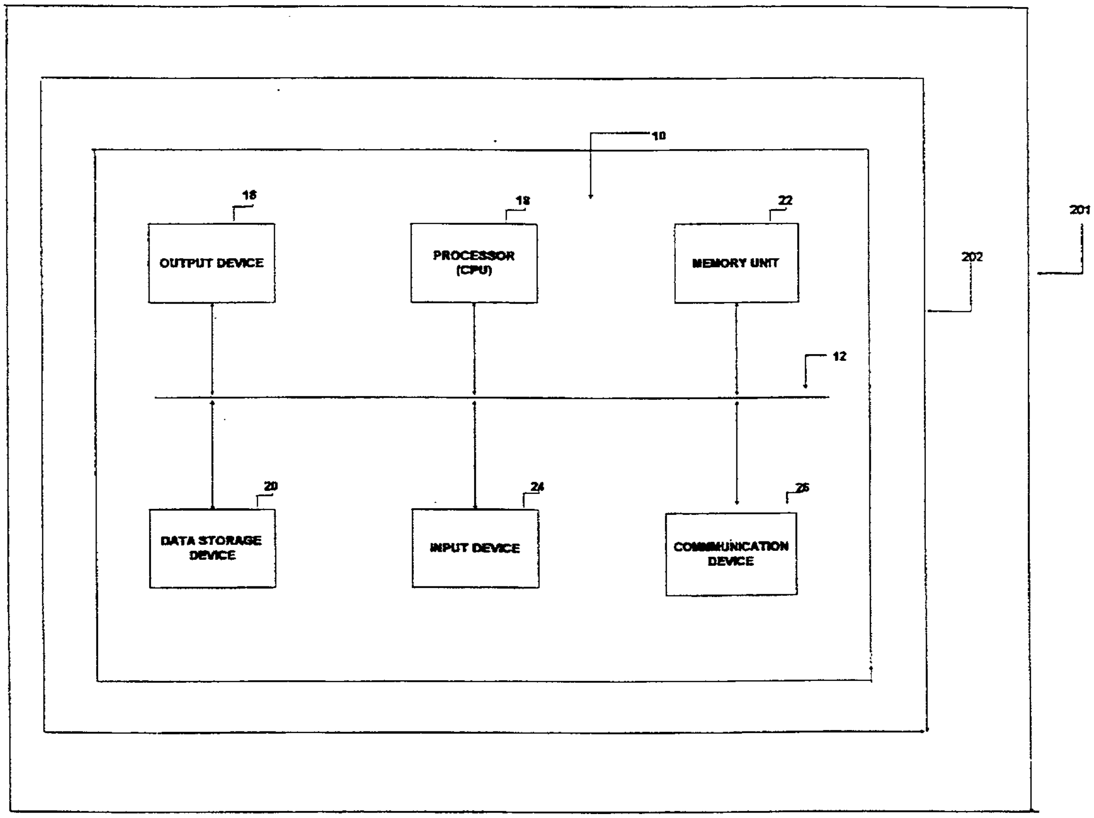

10
16
OUTPUT DEVICE
18
PROCESSOR (CPU)
22
MEMORY UNIT
201
202
12
20
DATA STORAGE DEVICE
24
INPUT DEVICE
26
COMMUNICATION DEVICE

FIG. 1

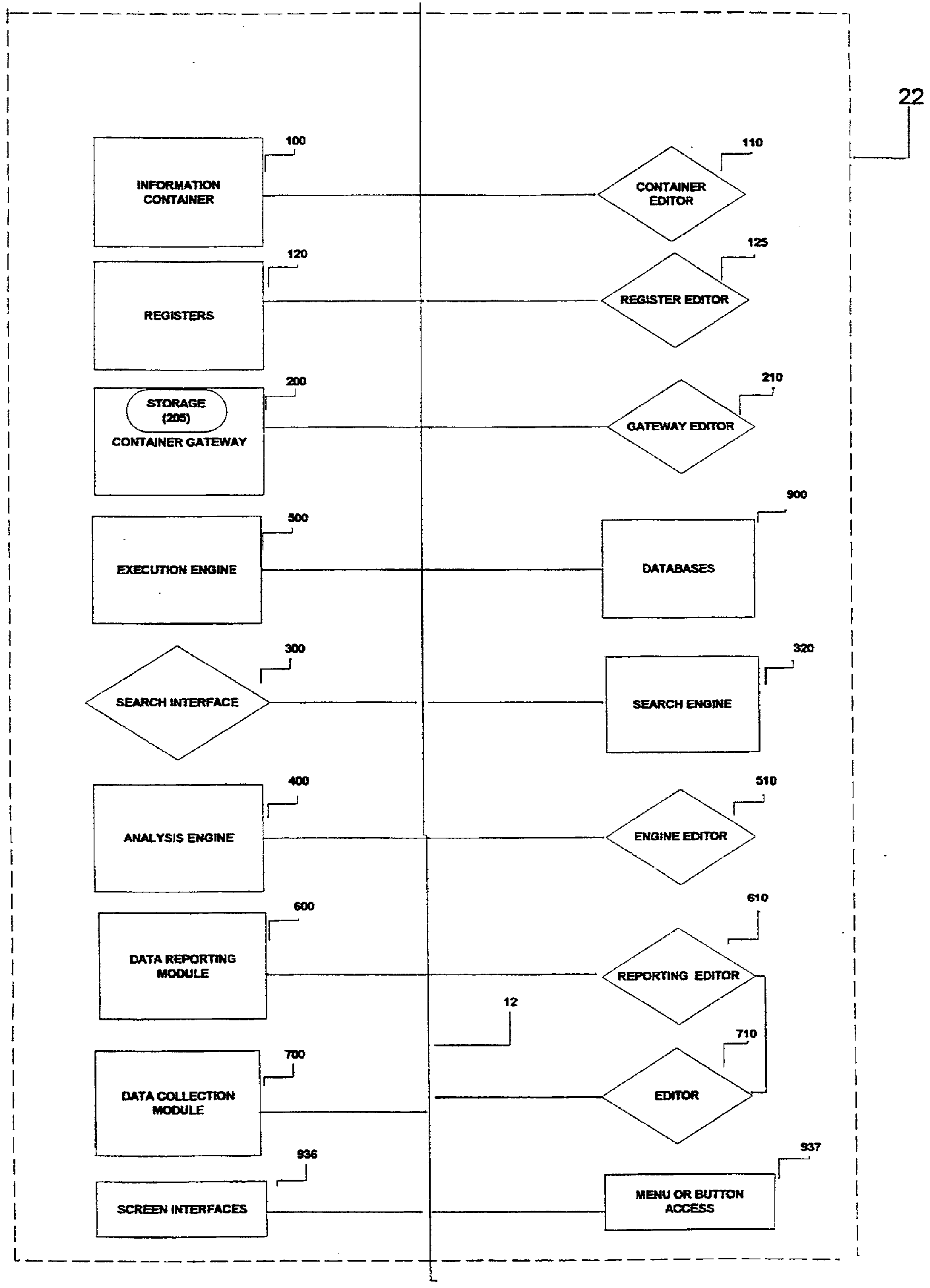

22
100
INFORMATION CONTAINER
110
CONTAINER EDITOR
120
REGISTERS
125
REGISTER EDITOR
200
STORAGE (205)
CONTAINER GATEWAY
210
GATEWAY EDITOR
500
EXECUTION ENGINE
900
DATABASES
300
SEARCH INTERFACE
320
SEARCH ENGINE
400
ANALYSIS ENGINE
510
ENGINE EDITOR
600
DATA REPORTING MODULE
610
REPORTING EDITOR
12
700
DATA COLLECTION MODULE
710
EDITOR
936
SCREEN INTERFACES
937
MENU OR BUTTON ACCESS

FIG. 2 A

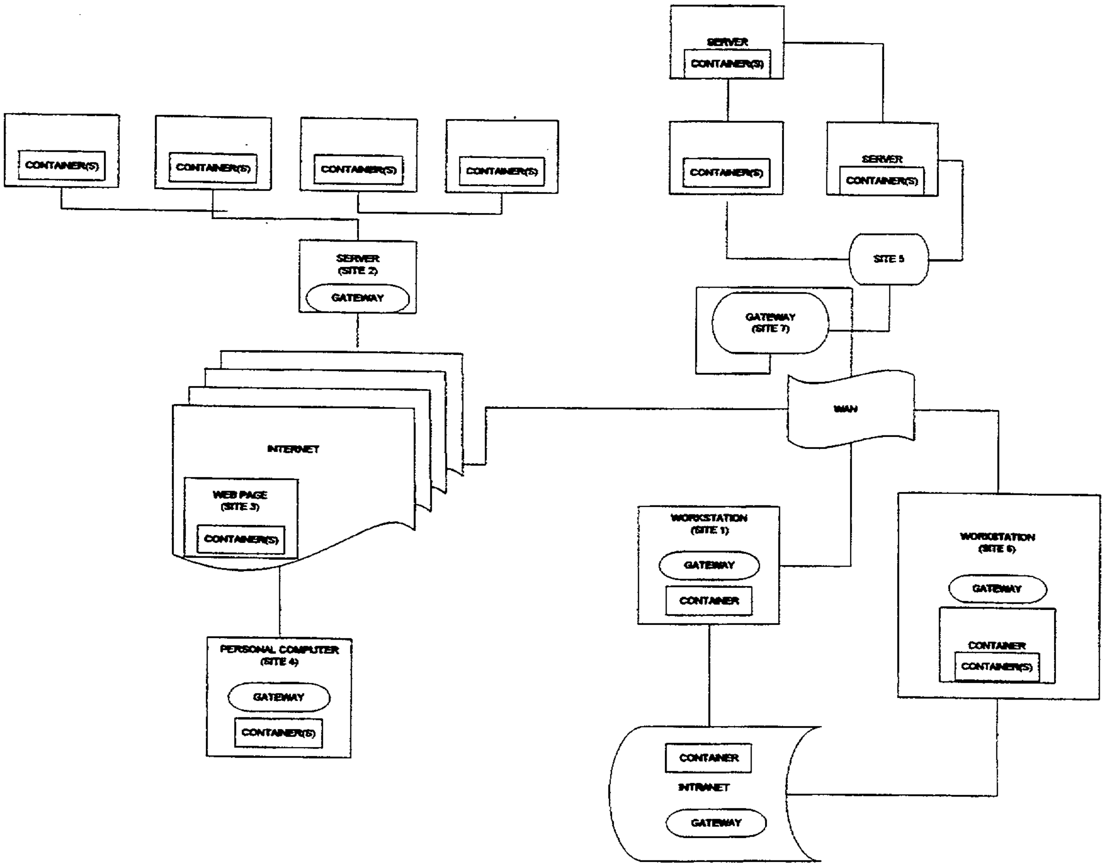

SERVER
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
SERVER
CONTAINER(S)
SERVER
(SITE 2)
GATEWAY
SITE 5
GATEWAY
(SITE 7)
INTERNET
WEB PAGE
(SITE 3)
CONTAINER(S)
WAN
WORKSTATION
(SITE 1)
GATEWAY
CONTAINER
WORKSTATION
(SITE 6)
GATEWAY
CONTAINER
CONTAINER(S)
PERSONAL COMPUTER
(SITE 4)
GATEWAY
CONTAINER(S)
CONTAINER
INTRANET
GATEWAY

FIG. 2 B

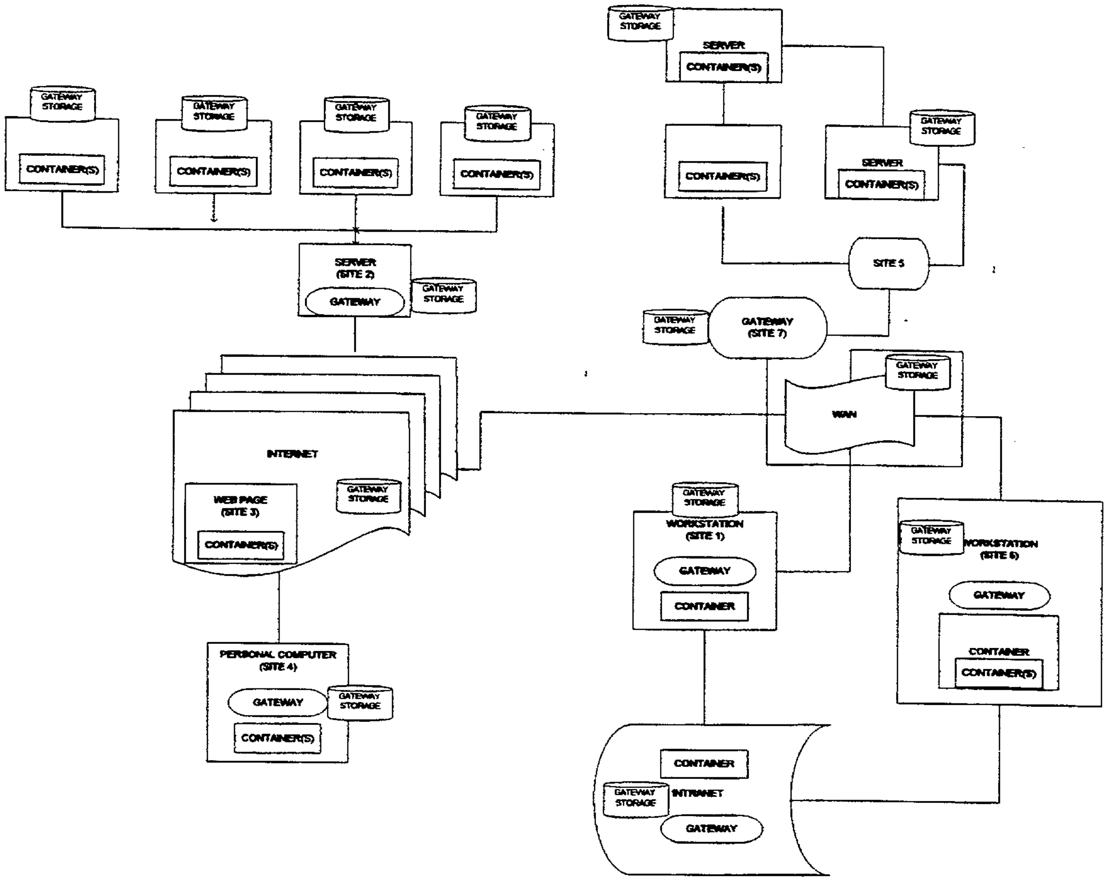

GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
CONTAINER(S)
SERVER (SITE 2)
GATEWAY
GATEWAY STORAGE
INTERNET
WEB PAGE (SITE 3)
CONTAINER(S)
GATEWAY STORAGE
PERSONAL COMPUTER (SITE 4)
GATEWAY
GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
SERVER
CONTAINER(S)
CONTAINER(S)
GATEWAY STORAGE
SERVER
CONTAINER(S)
SITE 5
GATEWAY STORAGE
GATEWAY (SITE 7)
GATEWAY STORAGE
WAN
GATEWAY STORAGE
WORKSTATION (SITE 1)
GATEWAY
CONTAINER
GATEWAY STORAGE
WORKSTATION (SITE 6)
GATEWAY
CONTAINER
CONTAINER(S)
CONTAINER
GATEWAY STORAGE
INTRANET
GATEWAY

FIG. 2 B 1

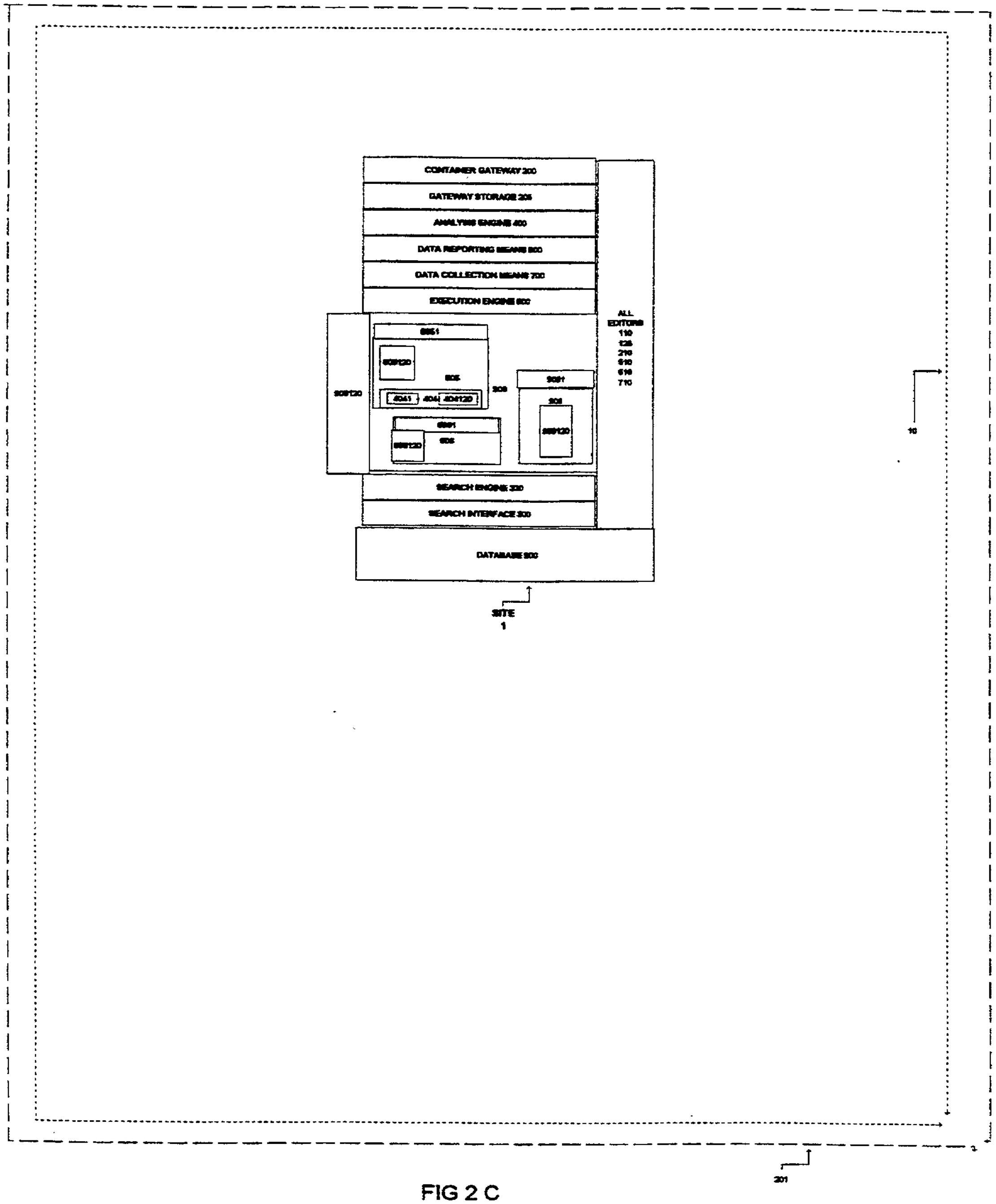

CONTAINER GATEWAY 200
GATEWAY STORAGE 205
ANALYSIS ENGINE 400
DATA REPORTING MEANS 500
DATA COLLECTION MEANS 700
EXECUTION ENGINE 800
809120
8051
805120
805
4041
404
404120
808
9081
808
809120
8061
806120
806
SEARCH ENGINE 320
SEARCH INTERFACE 300
ALL
EDITORS
110
125
210
510
610
710
DATABASE 900
SITE
1
10
201

FIG 2 C

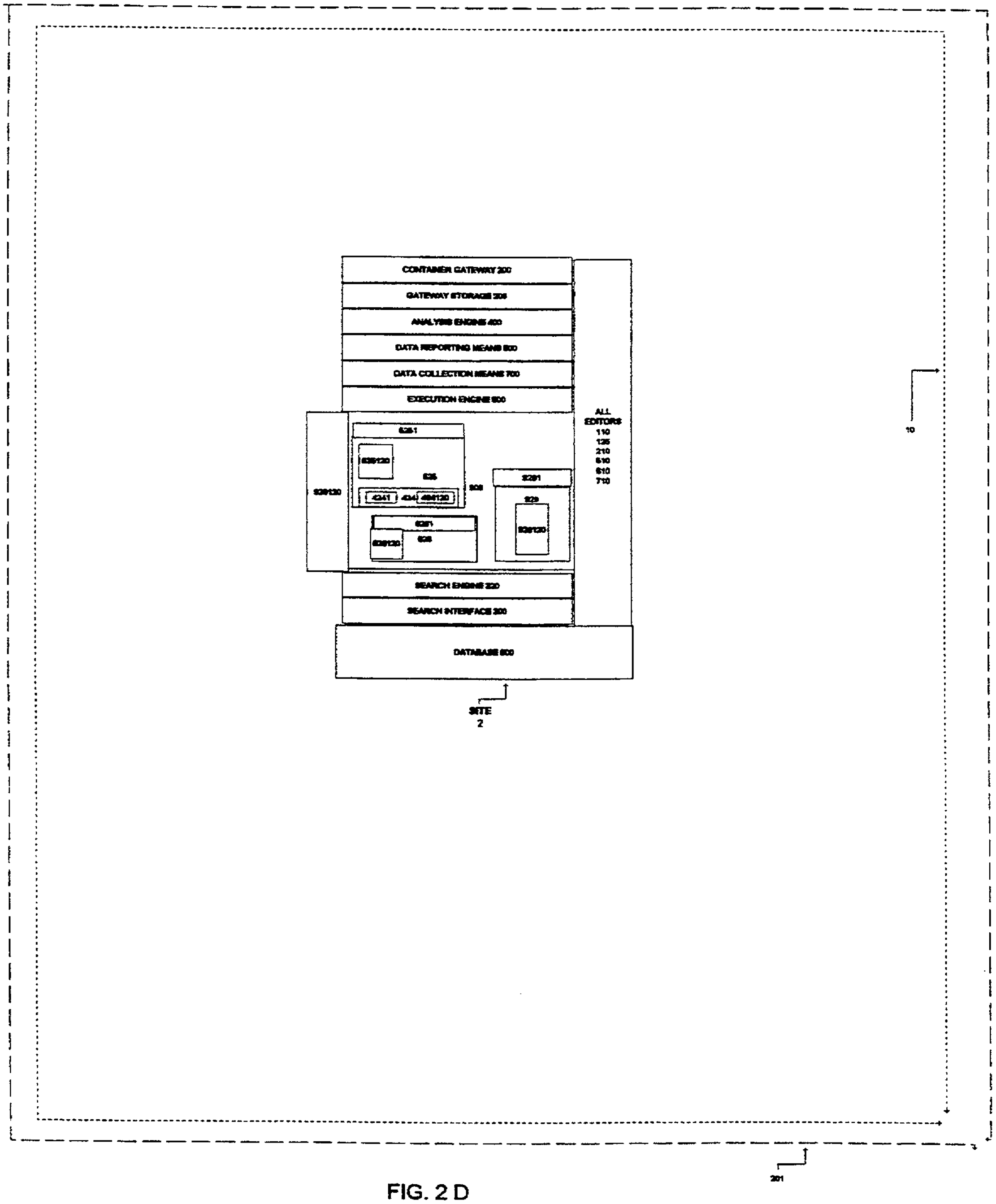

CONTAINER GATEWAY 200
GATEWAY STORAGE 205
ANALYSIS ENGINE 400
DATA REPORTING MEANS 500
DATA COLLECTION MEANS 700
EXECUTION ENGINE 800
ALL EDITORS
110
125
210
510
610
710
8281
828/120
828
4241
424
424/120
808
828/120
8281
828/120
828
8291
829
829/120
SEARCH ENGINE 220
SEARCH INTERFACE 300
DATABASE 600
SITE 2
10
201

FIG. 2 D

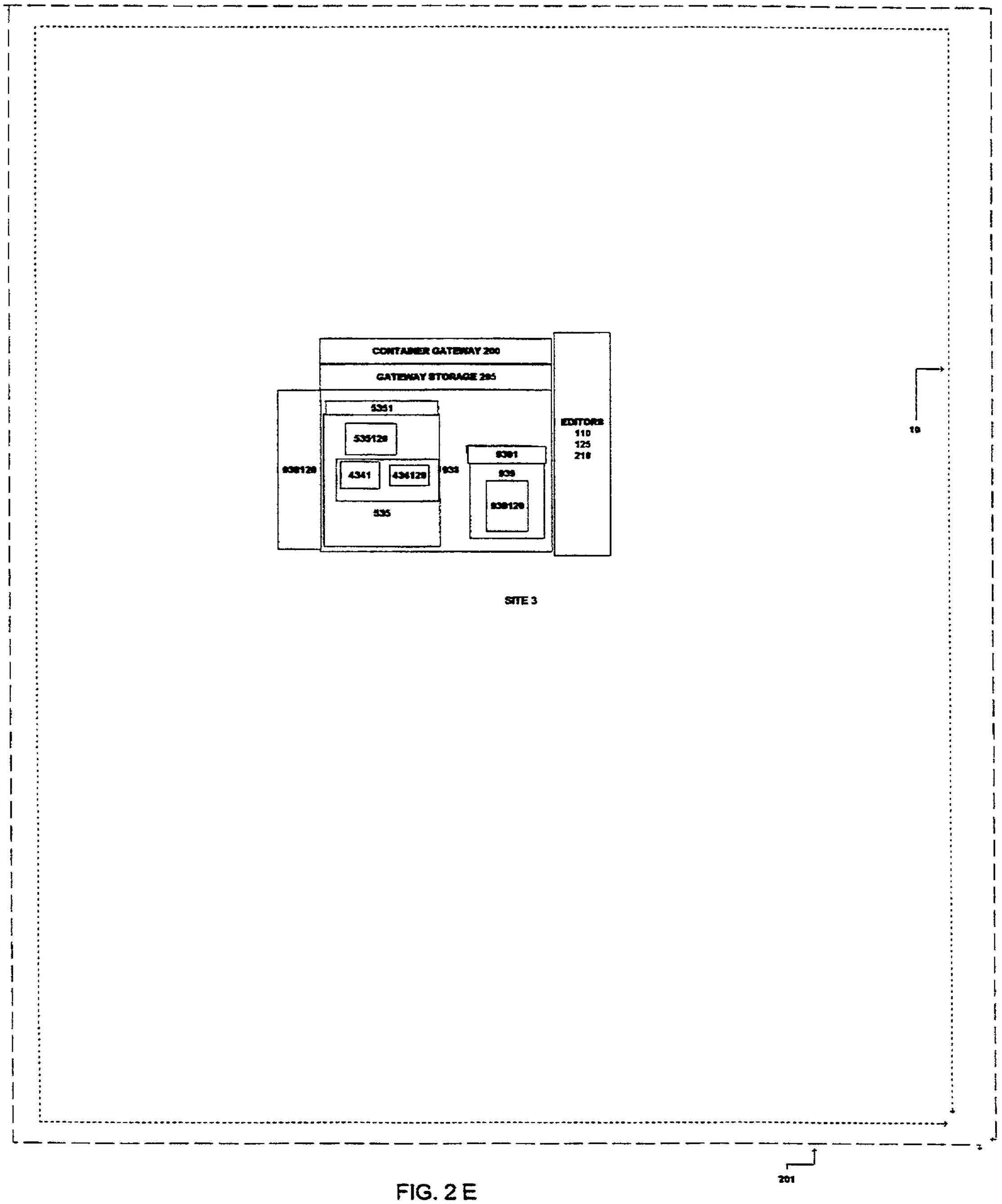

CONTAINER GATEWAY 200
GATEWAY STORAGE 295
5351
535120
4341
434120
535
930120
934
9301
930
930120
EDITORS
110
125
210
SITE 3
10
201

FIG. 2 E

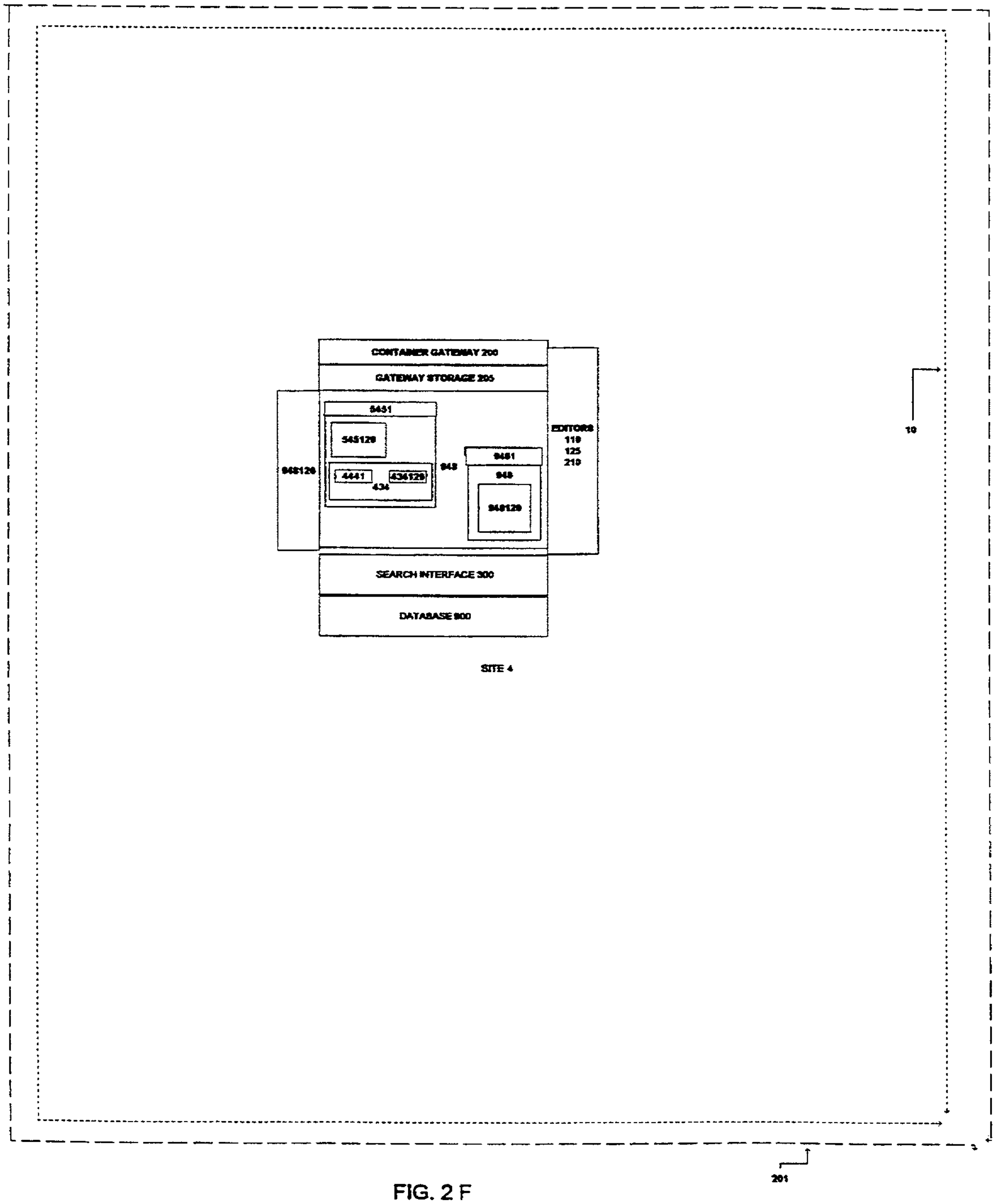

CONTAINER GATEWAY 200
GATEWAY STORAGE 205
5451
545120
948120
4441
436120
434
948
9481
948
948120
EDITORS
110
125
210
SEARCH INTERFACE 300
DATABASE 800
SITE 4
10
201

FIG. 2 F

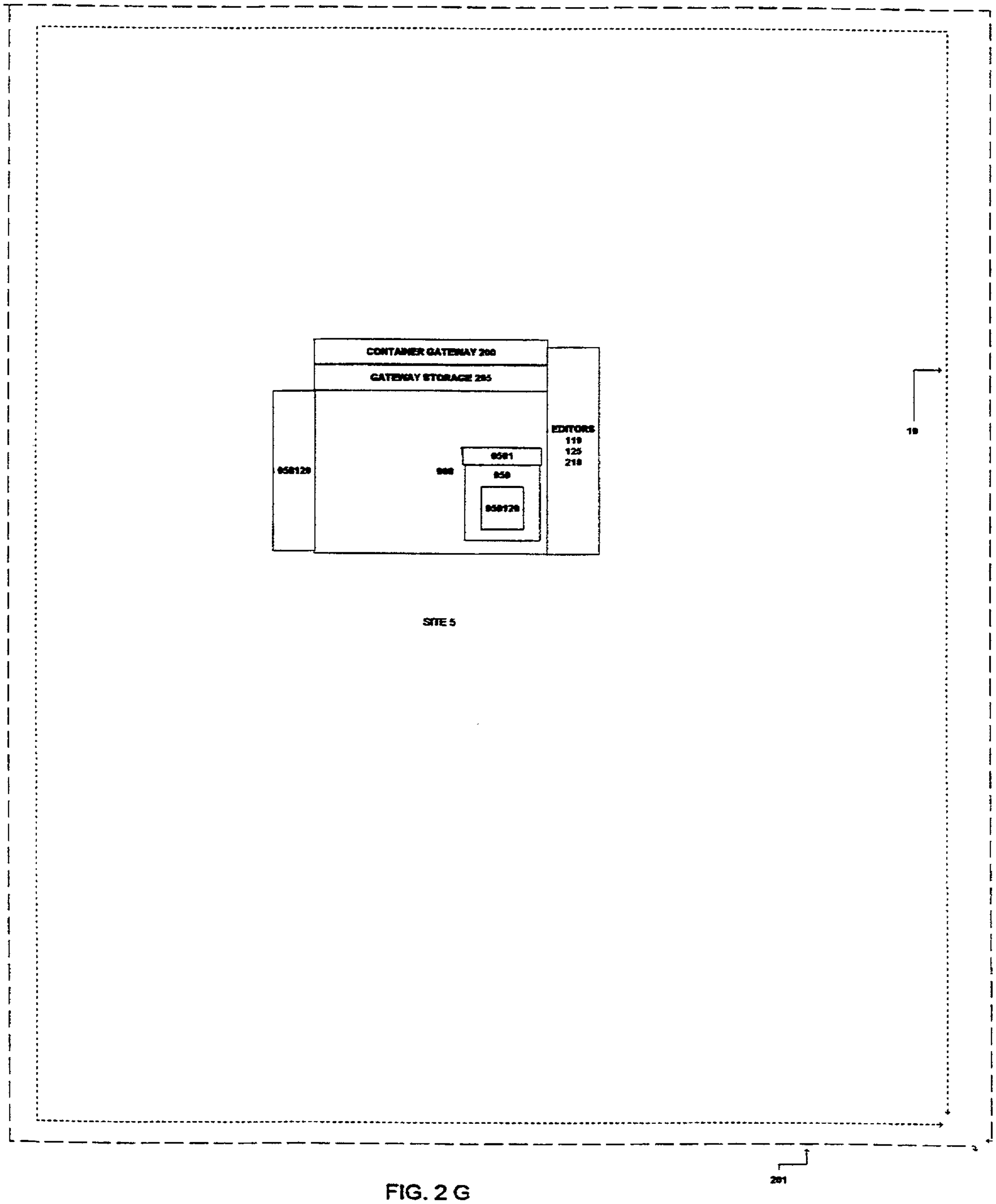

CONTAINER GATEWAY 200
GATEWAY STORAGE 295
950120
900
9501
950
950120
EDITORS
119
125
218
SITE 5
10
201

FIG. 2 G

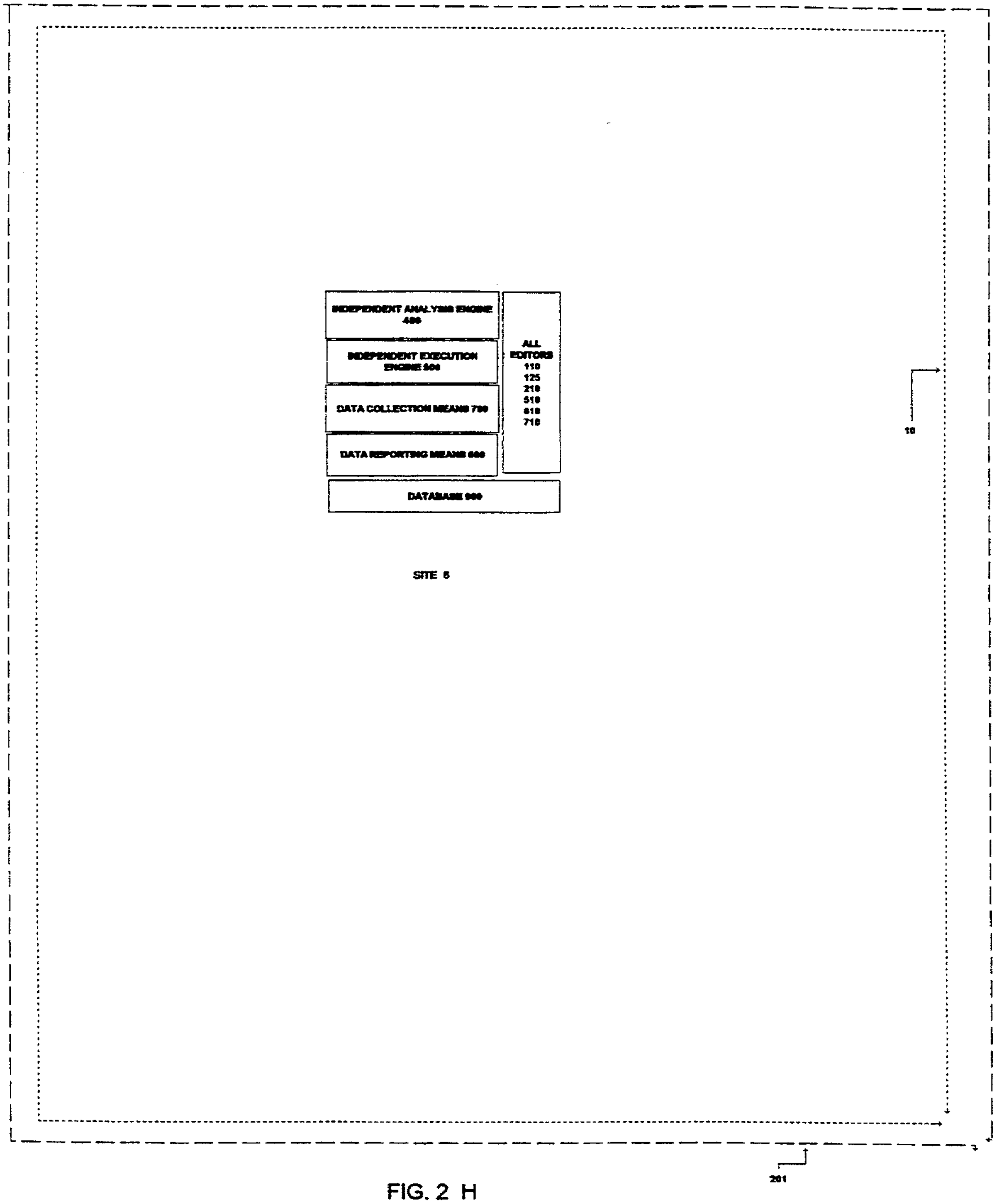

INDEPENDENT ANALYSIS ENGINE 400
INDEPENDENT EXECUTION ENGINE 500
DATA COLLECTION MEANS 700
DATA REPORTING MEANS 600
ALL EDITORS 110 125 210 510 610 710
DATABASE 900
SITE 6
10
201

FIG. 2 H

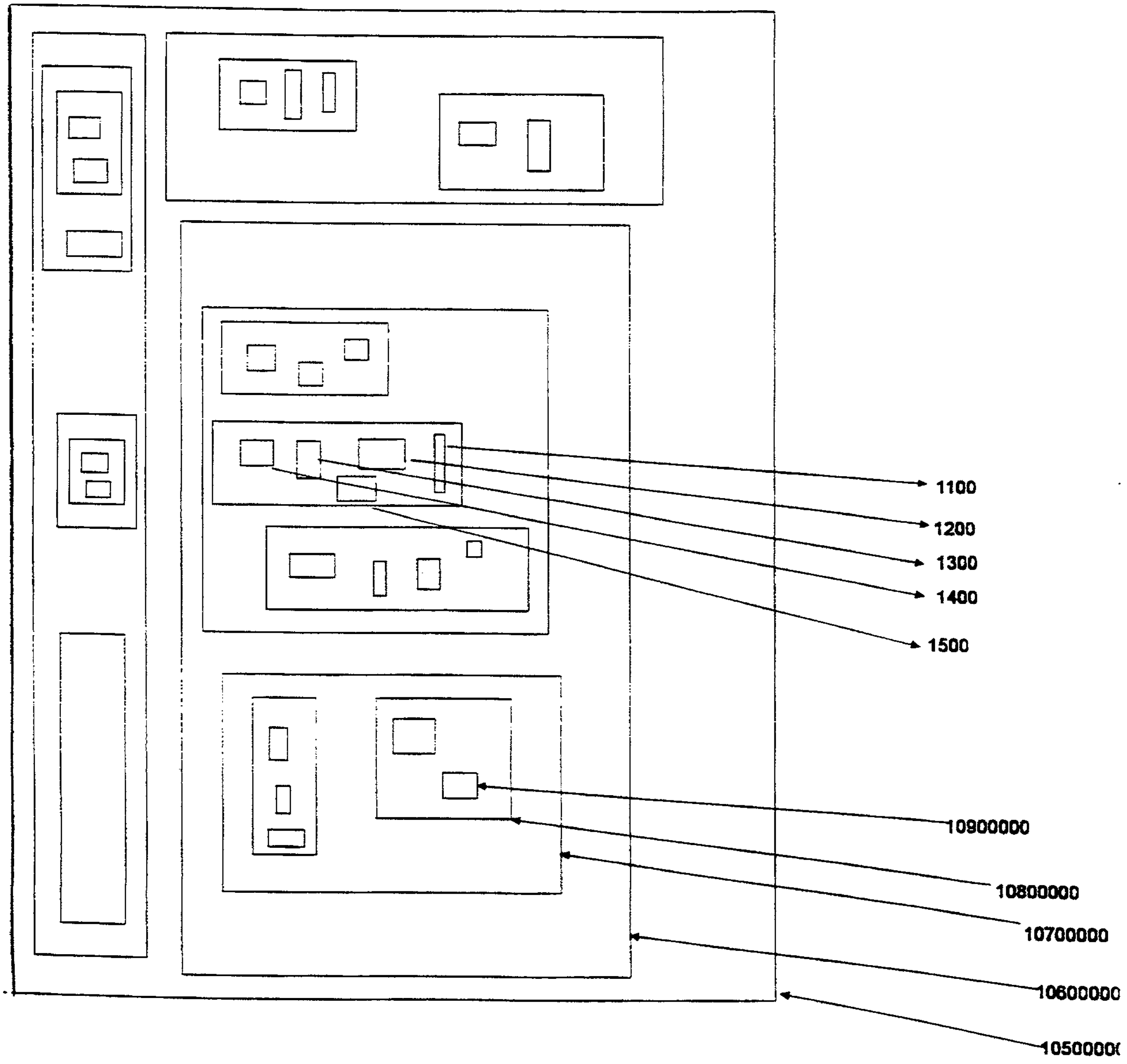

1100
1200
1300
1400
1500
10900000
10800000
10700000
10600000
1050000(

FIG. 3 A

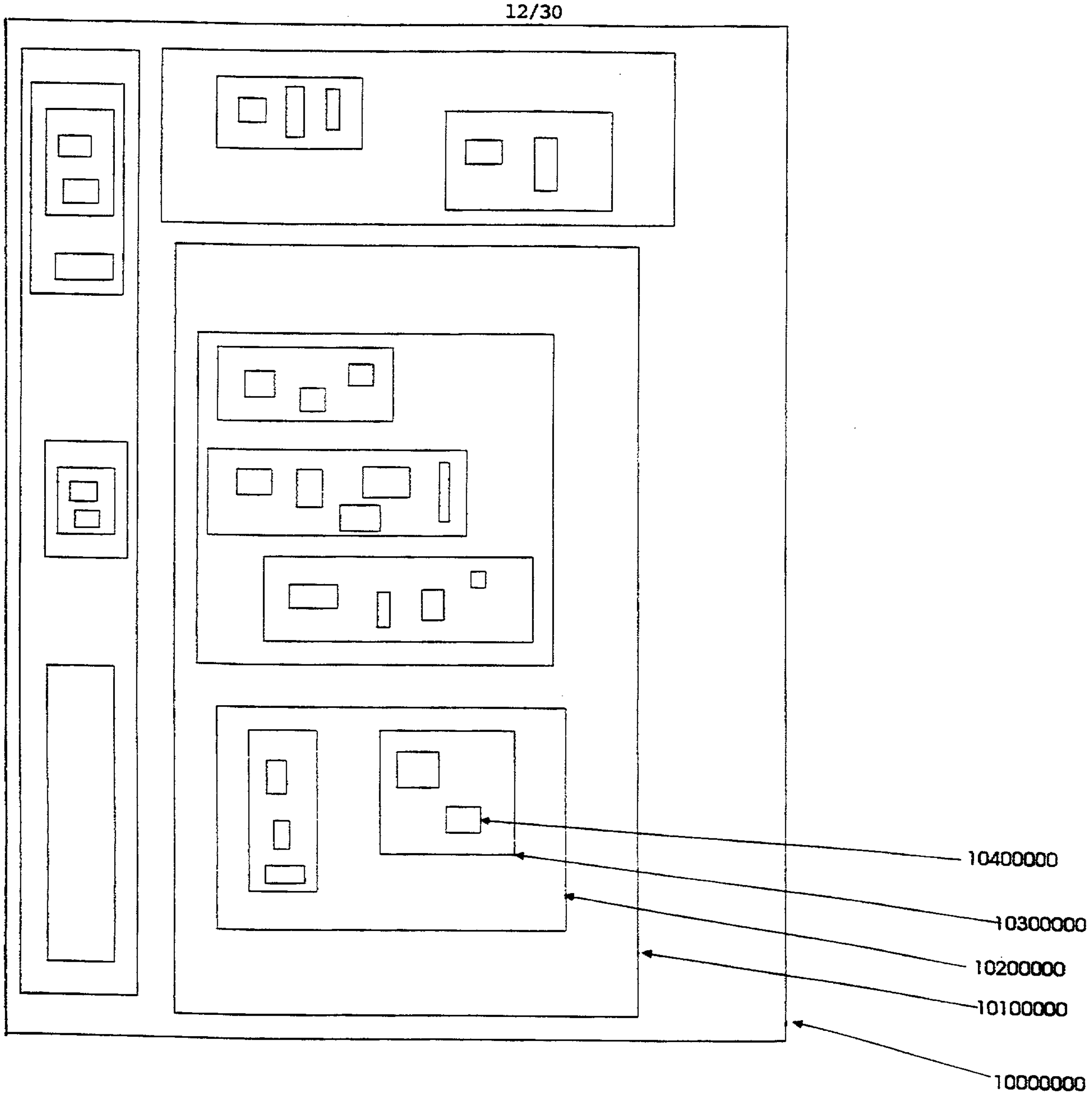


FIG. 3 B

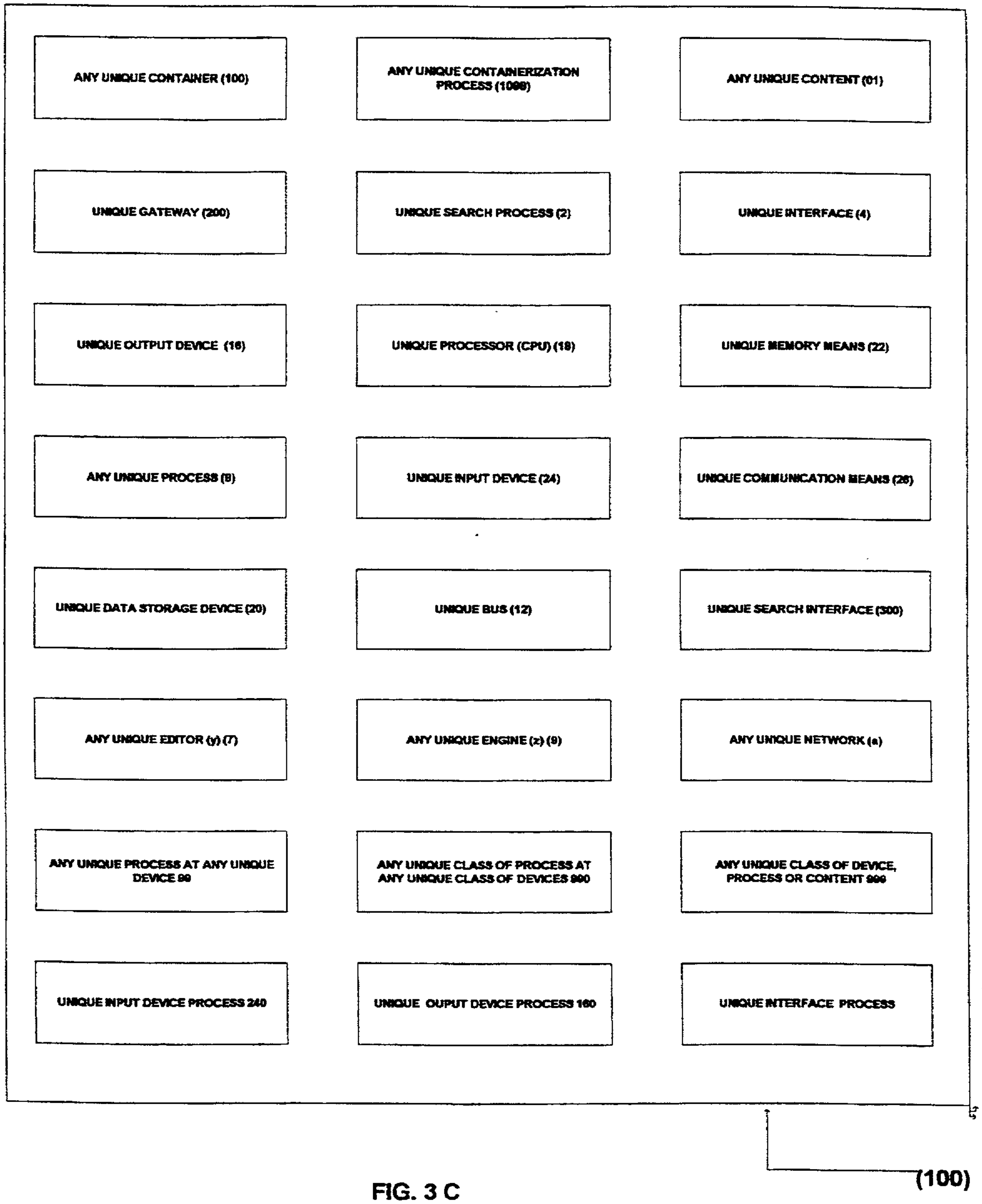

ANY UNIQUE CONTAINER (100)
ANY UNIQUE CONTAINERIZATION PROCESS (1000)
ANY UNIQUE CONTENT (01)
UNIQUE GATEWAY (200)
UNIQUE SEARCH PROCESS (2)
UNIQUE INTERFACE (4)
UNIQUE OUTPUT DEVICE (16)
UNIQUE PROCESSOR (CPU) (18)
UNIQUE MEMORY MEANS (22)
ANY UNIQUE PROCESS (8)
UNIQUE INPUT DEVICE (24)
UNIQUE COMMUNICATION MEANS (26)
UNIQUE DATA STORAGE DEVICE (20)
UNIQUE BUS (12)
UNIQUE SEARCH INTERFACE (300)
ANY UNIQUE EDITOR (y) (7)
ANY UNIQUE ENGINE (z) (9)
ANY UNIQUE NETWORK (a)
ANY UNIQUE PROCESS AT ANY UNIQUE DEVICE 99
ANY UNIQUE CLASS OF PROCESS AT ANY UNIQUE CLASS OF DEVICES 990
ANY UNIQUE CLASS OF DEVICE, PROCESS OR CONTENT 999
UNIQUE INPUT DEVICE PROCESS 240
UNIQUE OUPUT DEVICE PROCESS 160
UNIQUE INTERFACE PROCESS
(100)

FIG. 3 C

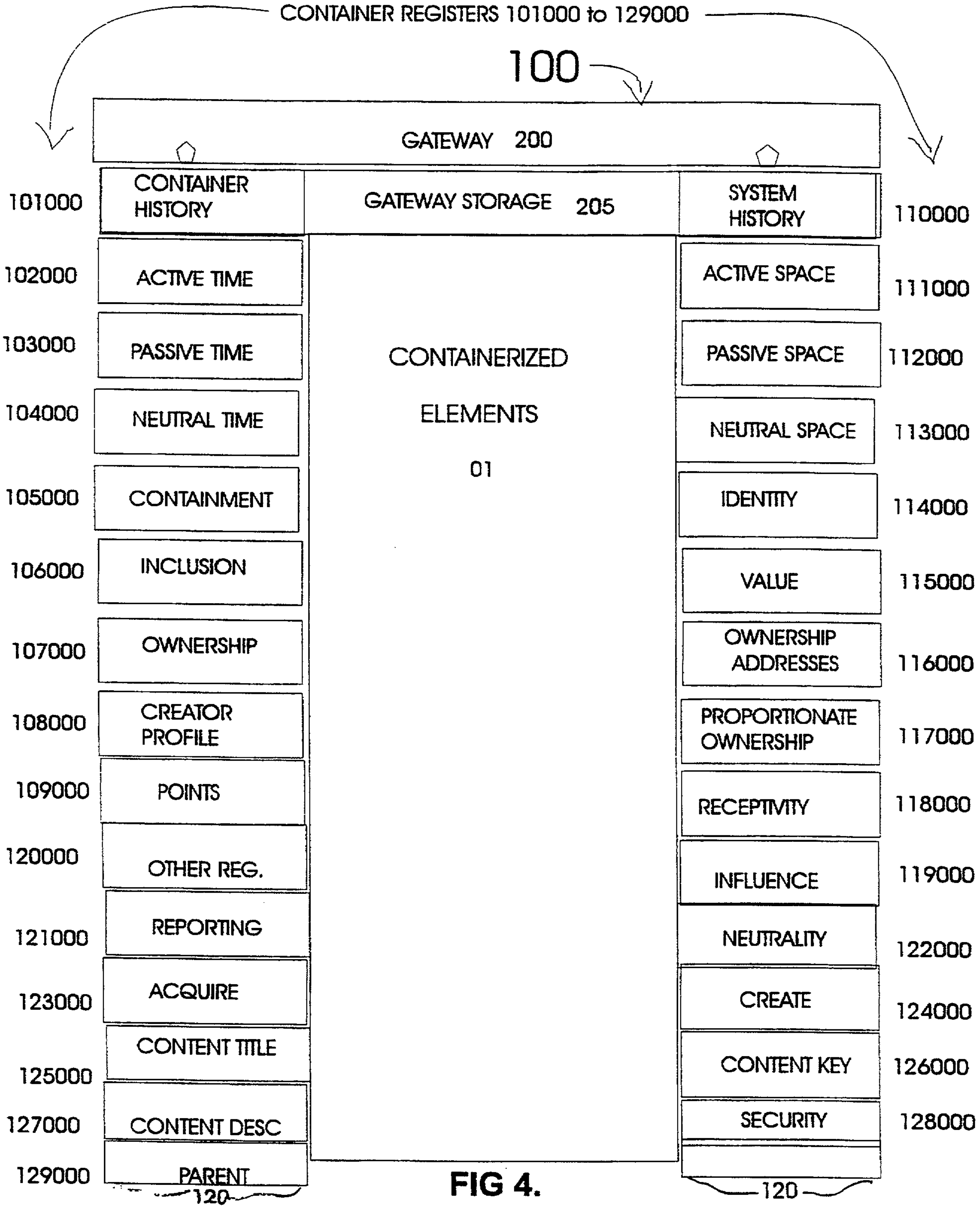

CONTAINER REGISTERS 101000 to 129000
100
GATEWAY 200
CONTAINER HISTORY
GATEWAY STORAGE 205
SYSTEM HISTORY
101000
110000
102000
ACTIVE TIME
ACTIVE SPACE
111000
103000
PASSIVE TIME
CONTAINERIZED ELEMENTS 01
PASSIVE SPACE
112000
104000
NEUTRAL TIME
NEUTRAL SPACE
113000
105000
CONTAINMENT
IDENTITY
114000
106000
INCLUSION
VALUE
115000
107000
OWNERSHIP
OWNERSHIP ADDRESSES
116000
108000
CREATOR PROFILE
PROPORTIONATE OWNERSHIP
117000
109000
POINTS
RECEPTIVITY
118000
120000
OTHER REG.
INFLUENCE
119000
121000
REPORTING
NEUTRALITY
122000
123000
ACQUIRE
CREATE
124000
125000
CONTENT TITLE
CONTENT KEY
126000
127000
CONTENT DESC
SECURITY
128000
129000
PARENT
120
120

FIG 4.

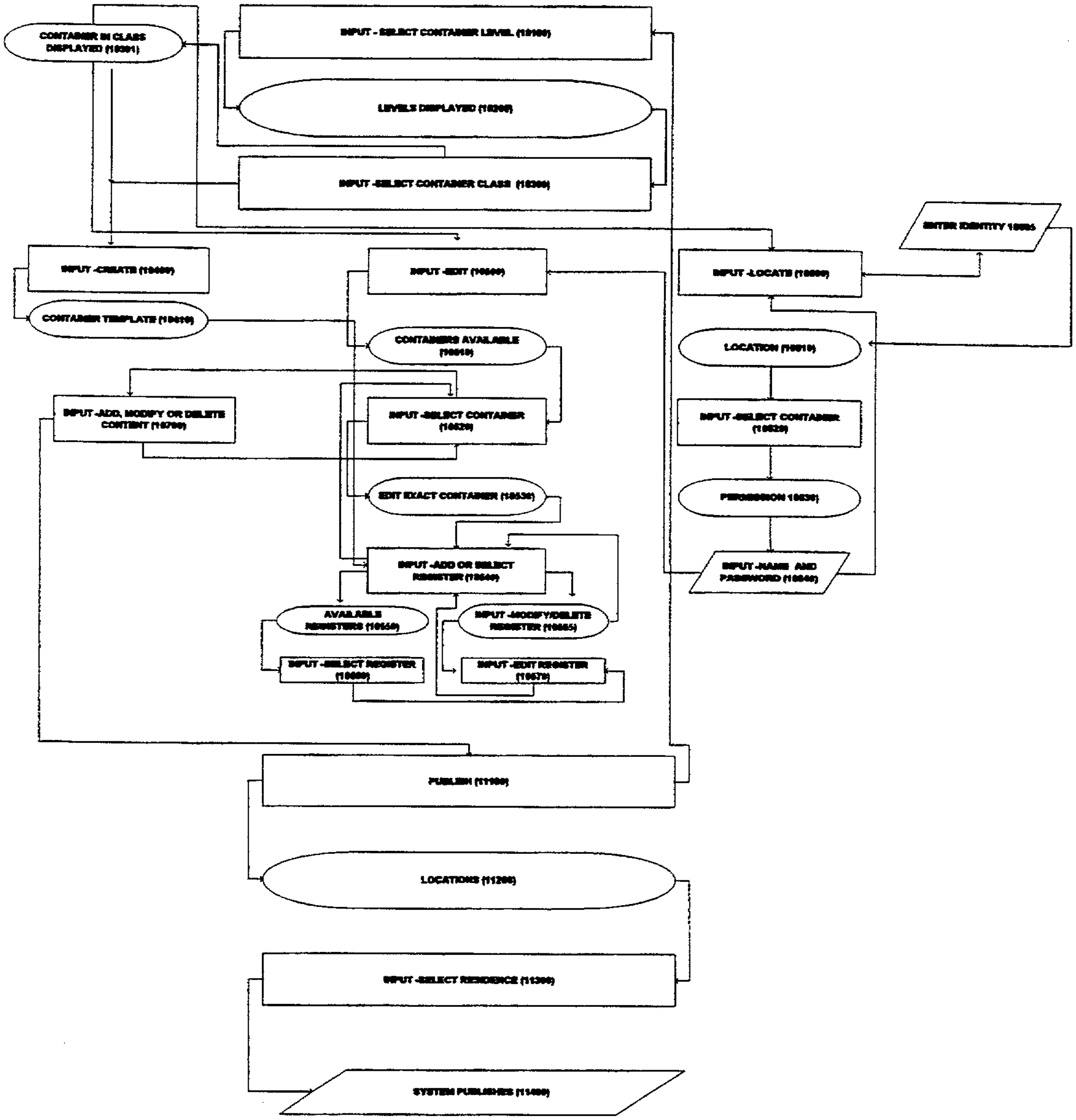

CONTAINER IN CLASS DISPLAYED (10301)
INPUT - SELECT CONTAINER LEVEL (10100)
LEVELS DISPLAYED (10200)
INPUT -SELECT CONTAINER CLASS (10300)
ENTER IDENTITY 10805
INPUT -CREATE (10400)
INPUT -EDIT (10500)
INPUT -LOCATE (10800)
CONTAINER TEMPLATE (10410)
CONTAINERS AVAILABLE (10510)
LOCATION (10810)
INPUT -ADD, MODIFY OR DELETE CONTENT (10700)
INPUT -SELECT CONTAINER (10520)
INPUT -SELECT CONTAINER (10820)
EDIT EXACT CONTAINER (10530)
PERMISSION 10830)
INPUT -ADD OR SELECT REGISTER (10540)
INPUT -NAME AND PASSWORD (10840)
AVAILABLE REGISTERS (10550)
INPUT -MODIFY/DELETE REGISTER (10565)
INPUT -SELECT REGISTER (10560)
INPUT -EDIT REGISTER (10570)
PUBLISH (11100)
LOCATIONS (11200)
INPUT -SELECT RESIDENCE (11300)
SYSTEM PUBLISHES (11400)

FIG. 5

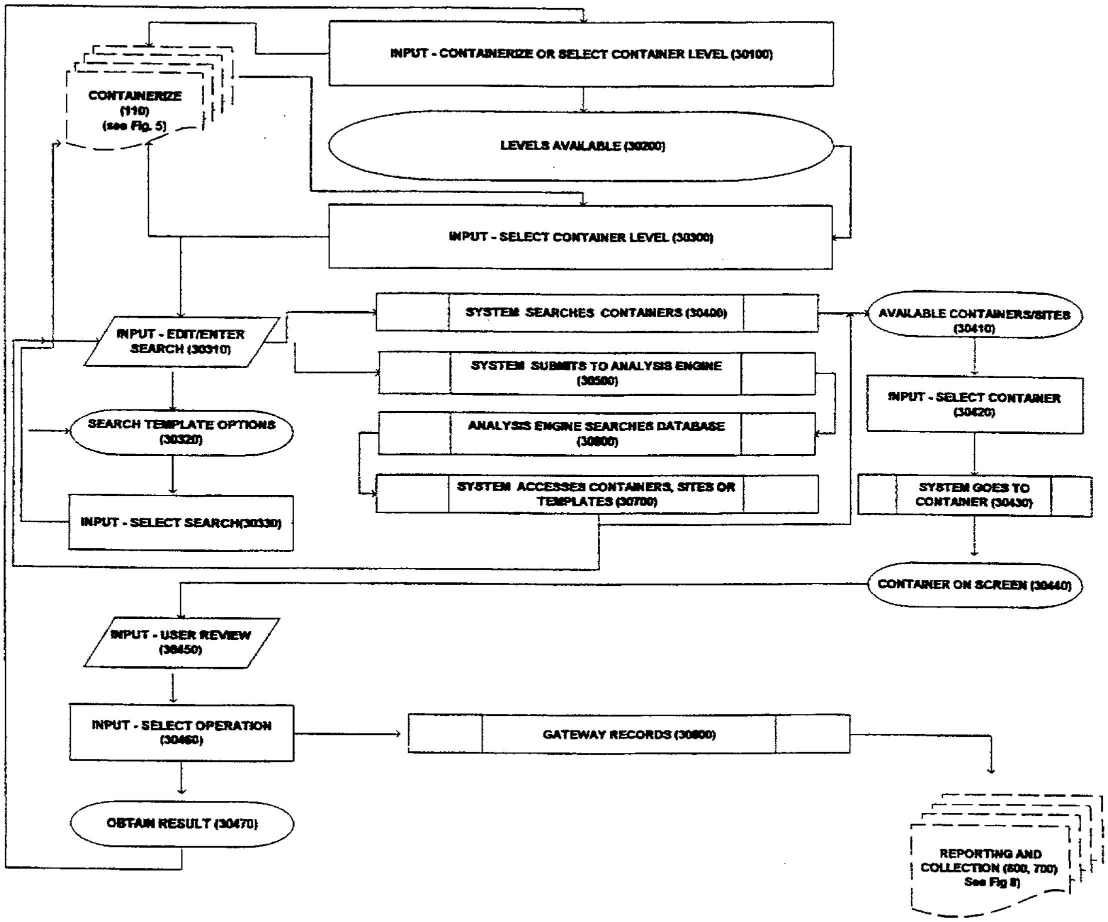

INPUT - CONTAINERIZE OR SELECT CONTAINER LEVEL (30100)
CONTAINERIZE (110) (see Fig. 5)
LEVELS AVAILABLE (30200)
INPUT - SELECT CONTAINER LEVEL (30300)
INPUT - EDIT/ENTER SEARCH (30310)
SYSTEM SEARCHES CONTAINERS (30400)
AVAILABLE CONTAINERS/SITES (30410)
SYSTEM SUBMITS TO ANALYSIS ENGINE (30500)
INPUT - SELECT CONTAINER (30420)
SEARCH TEMPLATE OPTIONS (30320)
ANALYSIS ENGINE SEARCHES DATABASE (30600)
SYSTEM ACCESSES CONTAINERS, SITES OR TEMPLATES (30700)
SYSTEM GOES TO CONTAINER (30430)
INPUT - SELECT SEARCH(30330)
CONTAINER ON SCREEN (30440)
INPUT - USER REVIEW (30450)
INPUT - SELECT OPERATION (30460)
GATEWAY RECORDS (30800)
OBTAIN RESULT (30470)
REPORTING AND COLLECTION (600, 700) See Fig 8)

FIG. 6

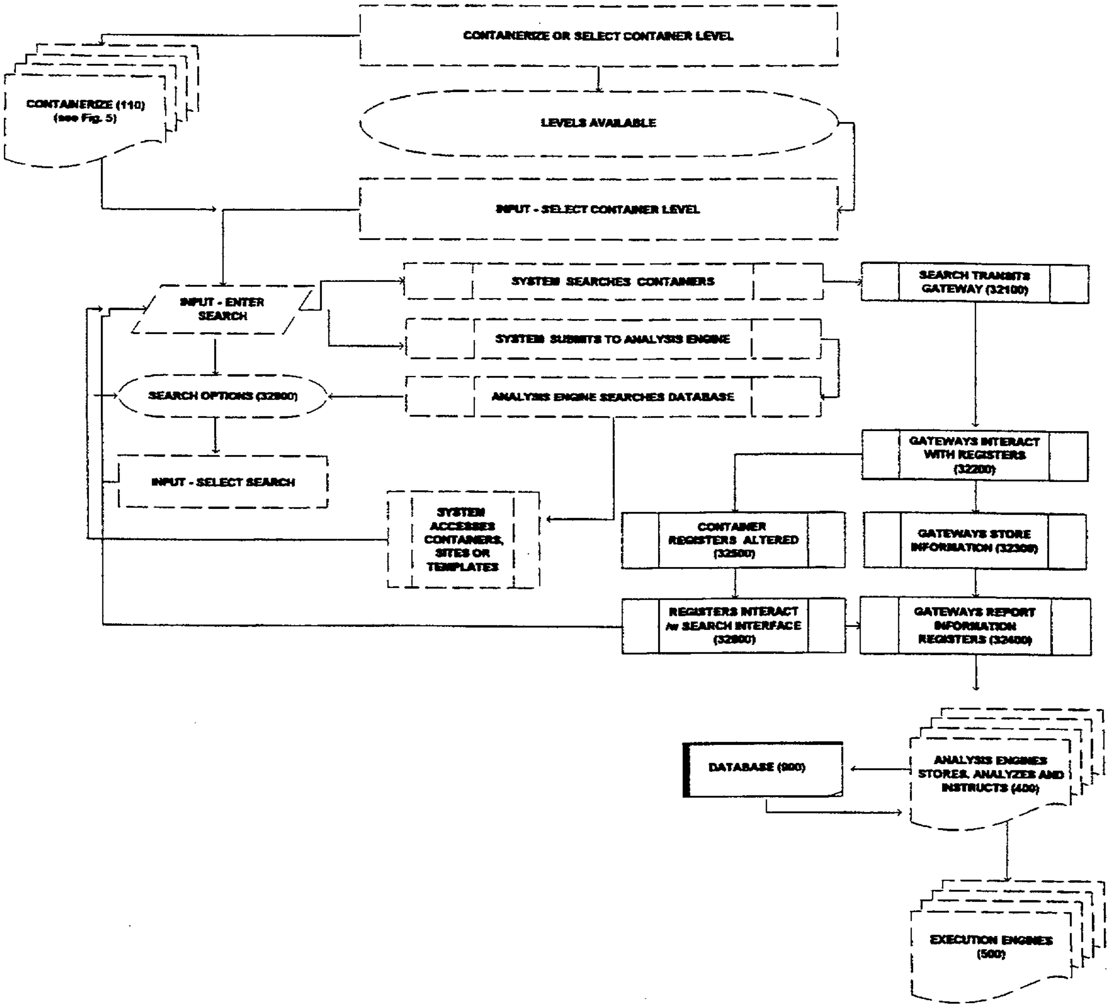

CONTAINERIZE OR SELECT CONTAINER LEVEL
CONTAINERIZE (110) (see Fig. 5)
LEVELS AVAILABLE
INPUT - SELECT CONTAINER LEVEL
INPUT - ENTER SEARCH
SYSTEM SEARCHES CONTAINERS
SEARCH TRANSITS GATEWAY (32100)
SYSTEM SUBMITS TO ANALYSIS ENGINE
SEARCH OPTIONS (32900)
ANALYSIS ENGINE SEARCHES DATABASE
INPUT - SELECT SEARCH
GATEWAYS INTERACT WITH REGISTERS (32200)
SYSTEM ACCESSES CONTAINERS, SITES OR TEMPLATES
CONTAINER REGISTERS ALTERED (32500)
GATEWAYS STORE INFORMATION (32300)
REGISTERS INTERACT /w SEARCH INTERFACE (32800)
GATEWAYS REPORT INFORMATION REGISTERS (32400)
DATABASE (900)
ANALYSIS ENGINES STORES, ANALYZES AND INSTRUCTS (400)
EXECUTION ENGINES (500)

FIG. 7

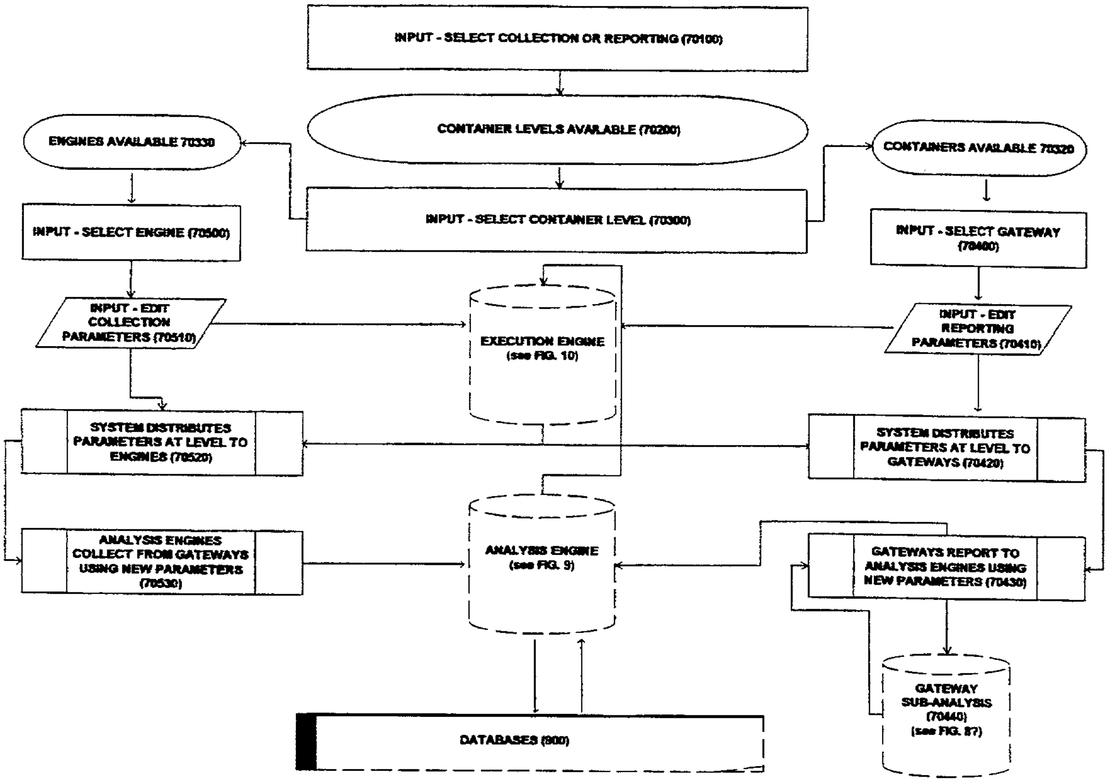

INPUT - SELECT COLLECTION OR REPORTING (70100)
CONTAINER LEVELS AVAILABLE (70200)
ENGINES AVAILABLE 70330
CONTAINERS AVAILABLE 70320
INPUT - SELECT CONTAINER LEVEL (70300)
INPUT - SELECT ENGINE (70500)
INPUT - SELECT GATEWAY (70400)
INPUT - EDIT COLLECTION PARAMETERS (70510)
EXECUTION ENGINE (see FIG. 10)
INPUT - EDIT REPORTING PARAMETERS (70410)
SYSTEM DISTRIBUTES PARAMETERS AT LEVEL TO ENGINES (70520)
SYSTEM DISTRIBUTES PARAMETERS AT LEVEL TO GATEWAYS (70420)
ANALYSIS ENGINES COLLECT FROM GATEWAYS USING NEW PARAMETERS (70530)
ANALYSIS ENGINE (see FIG. 9)
GATEWAYS REPORT TO ANALYSIS ENGINES USING NEW PARAMETERS (70430)
GATEWAY SUB-ANALYSIS (70440) (see FIG. 87)
DATABASES (900)

FIG. 8

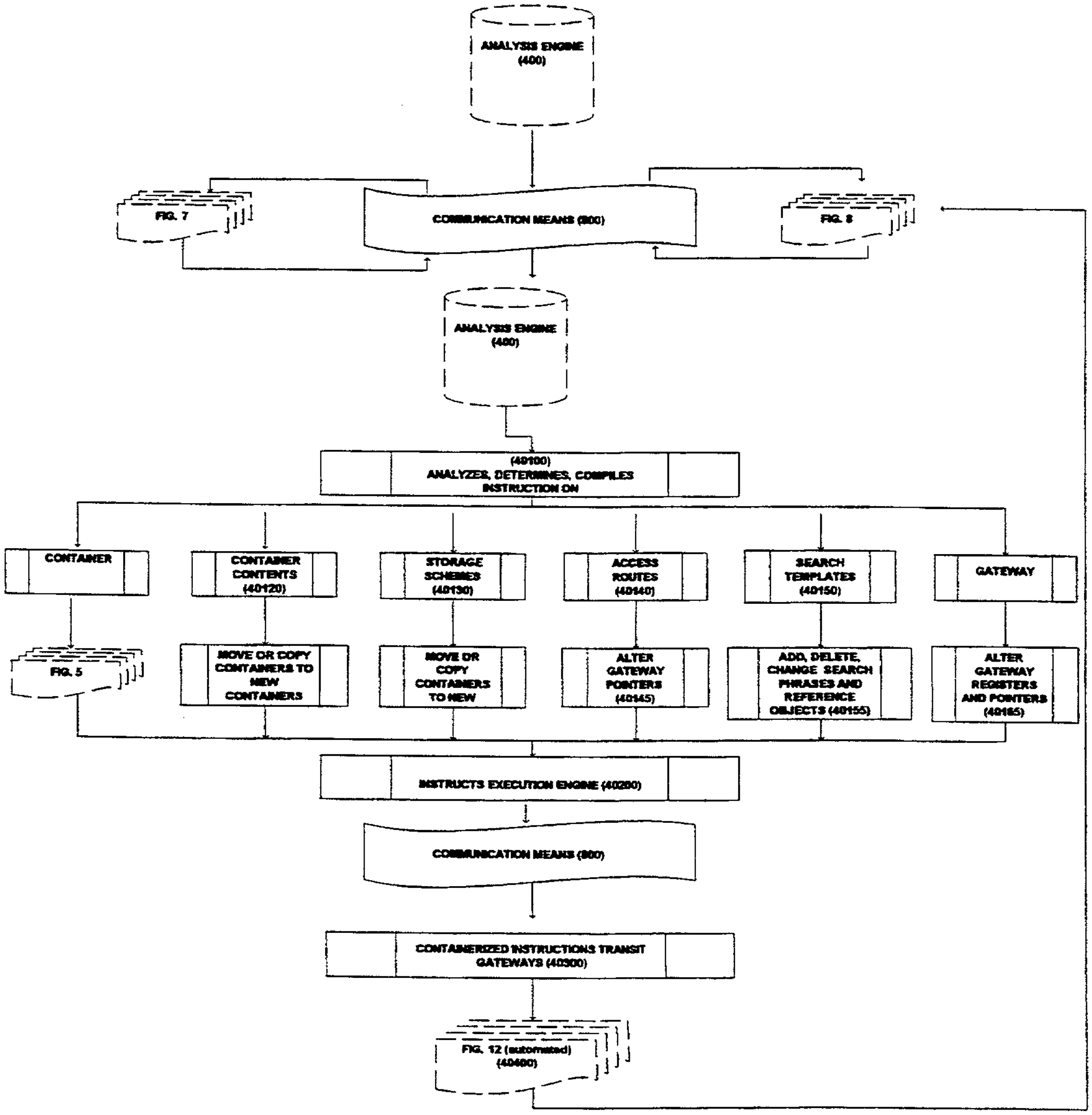

ANALYSIS ENGINE
(400)
FIG. 7
COMMUNICATION MEANS (800)
FIG. 8
ANALYSIS ENGINE
(400)
(40100)
ANALYZES, DETERMINES, COMPILES
INSTRUCTION ON
CONTAINER
CONTAINER
CONTENTS
(40120)
STORAGE
SCHEMES
(40130)
ACCESS
ROUTES
(40140)
SEARCH
TEMPLATES
(40150)
GATEWAY
FIG. 5
MOVE OR COPY
CONTAINERS TO
NEW
CONTAINERS
MOVE OR
COPY
CONTAINERS
TO NEW
ALTER
GATEWAY
POINTERS
(40145)
ADD, DELETE,
CHANGE SEARCH
PHRASES AND
REFERENCE
OBJECTS (40155)
ALTER
GATEWAY
REGISTERS
AND POINTERS
(40165)
INSTRUCTS EXECUTION ENGINE (40200)
COMMUNICATION MEANS (800)
CONTAINERIZED INSTRUCTIONS TRANSIT
GATEWAYS (40300)
FIG. 12 (automated)
(40400)

FIG. 9

EXECUTION ENGINE

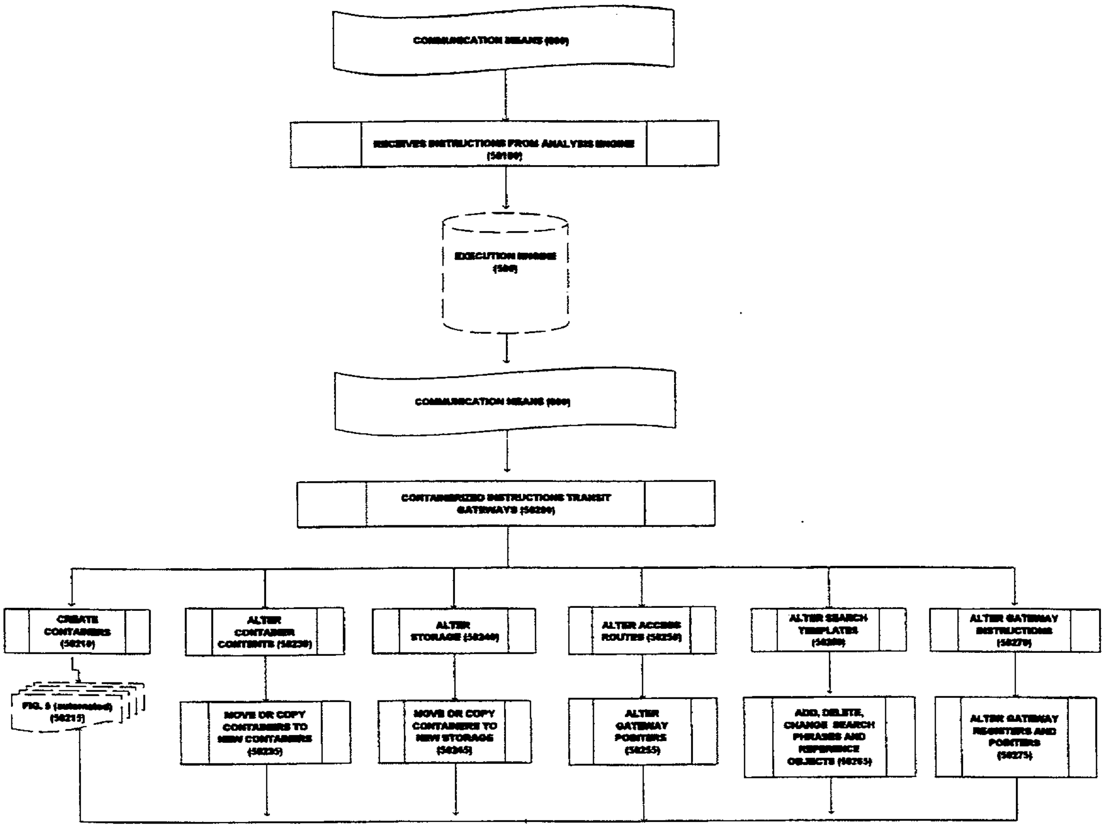

COMMUNICATION MEANS (800)
RECEIVES INSTRUCTIONS FROM ANALYSIS ENGINE (50190)
EXECUTION ENGINE (500)
COMMUNICATION MEANS (800)
CONTAINERIZED INSTRUCTIONS TRANSIT GATEWAYS (50200)
CREATE CONTAINERS (50210)
FIG. 8 (automated) (50215)
ALTER CONTAINER CONTENTS (50230)
MOVE OR COPY CONTAINERS TO NEW CONTAINERS (50235)
ALTER STORAGE (50240)
MOVE OR COPY CONTAINERS TO NEW STORAGE (50245)
ALTER ACCESS ROUTES (50250)
ALTER GATEWAY POINTERS (50255)
ALTER SEARCH TEMPLATES (50260)
ADD, DELETE, CHANGE SEARCH PHRASES AND REFERENCE OBJECTS (50265)
ALTER GATEWAY INSTRUCTIONS (50270)
ALTER GATEWAY REGISTERS AND POINTERS (50275)

FIG.10

GATEWAY EDITOR

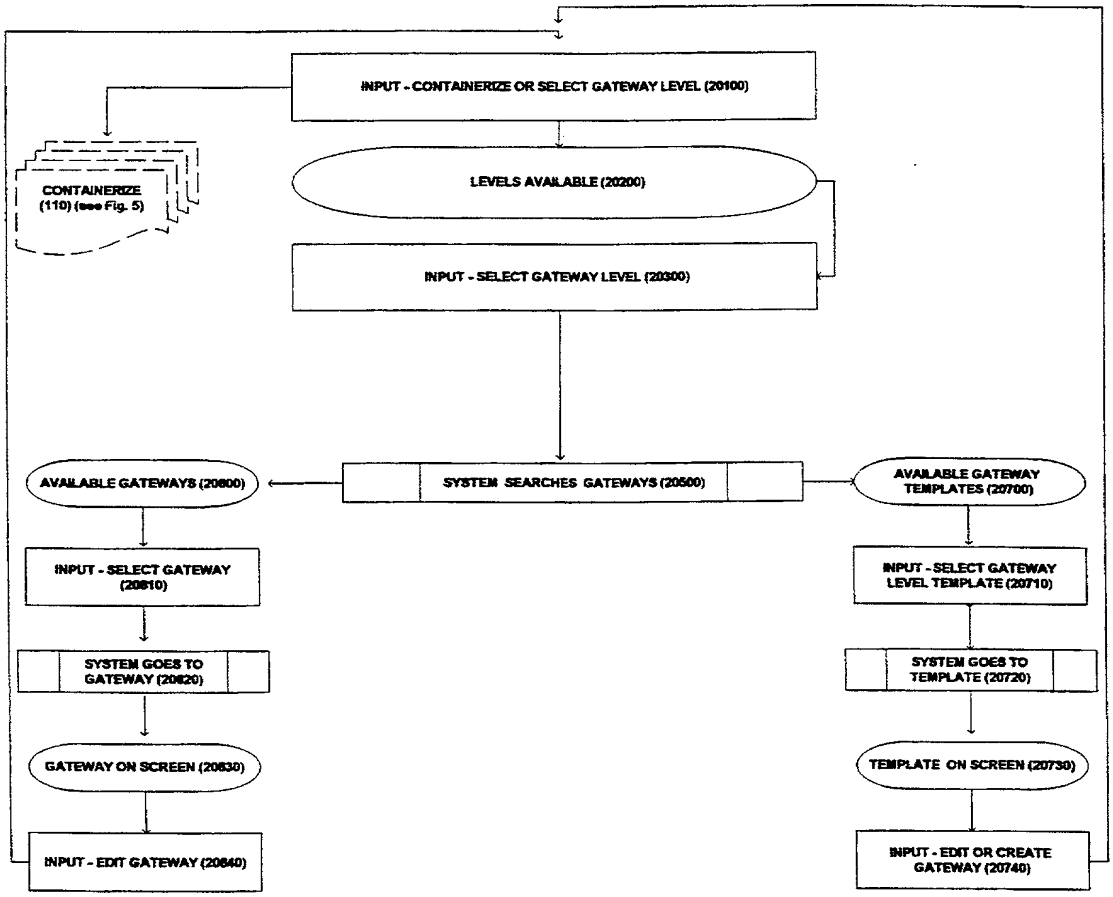

INPUT - CONTAINERIZE OR SELECT GATEWAY LEVEL (20100)
CONTAINERIZE (110) (see Fig. 5)
LEVELS AVAILABLE (20200)
INPUT - SELECT GATEWAY LEVEL (20300)
AVAILABLE GATEWAYS (20600)
SYSTEM SEARCHES GATEWAYS (20500)
AVAILABLE GATEWAY TEMPLATES (20700)
INPUT - SELECT GATEWAY (20610)
INPUT - SELECT GATEWAY LEVEL TEMPLATE (20710)
SYSTEM GOES TO GATEWAY (20620)
SYSTEM GOES TO TEMPLATE (20720)
GATEWAY ON SCREEN (20630)
TEMPLATE ON SCREEN (20730)
INPUT - EDIT GATEWAY (20640)
INPUT - EDIT OR CREATE GATEWAY (20740)

FIG. 11

GATEWAY PROCESS

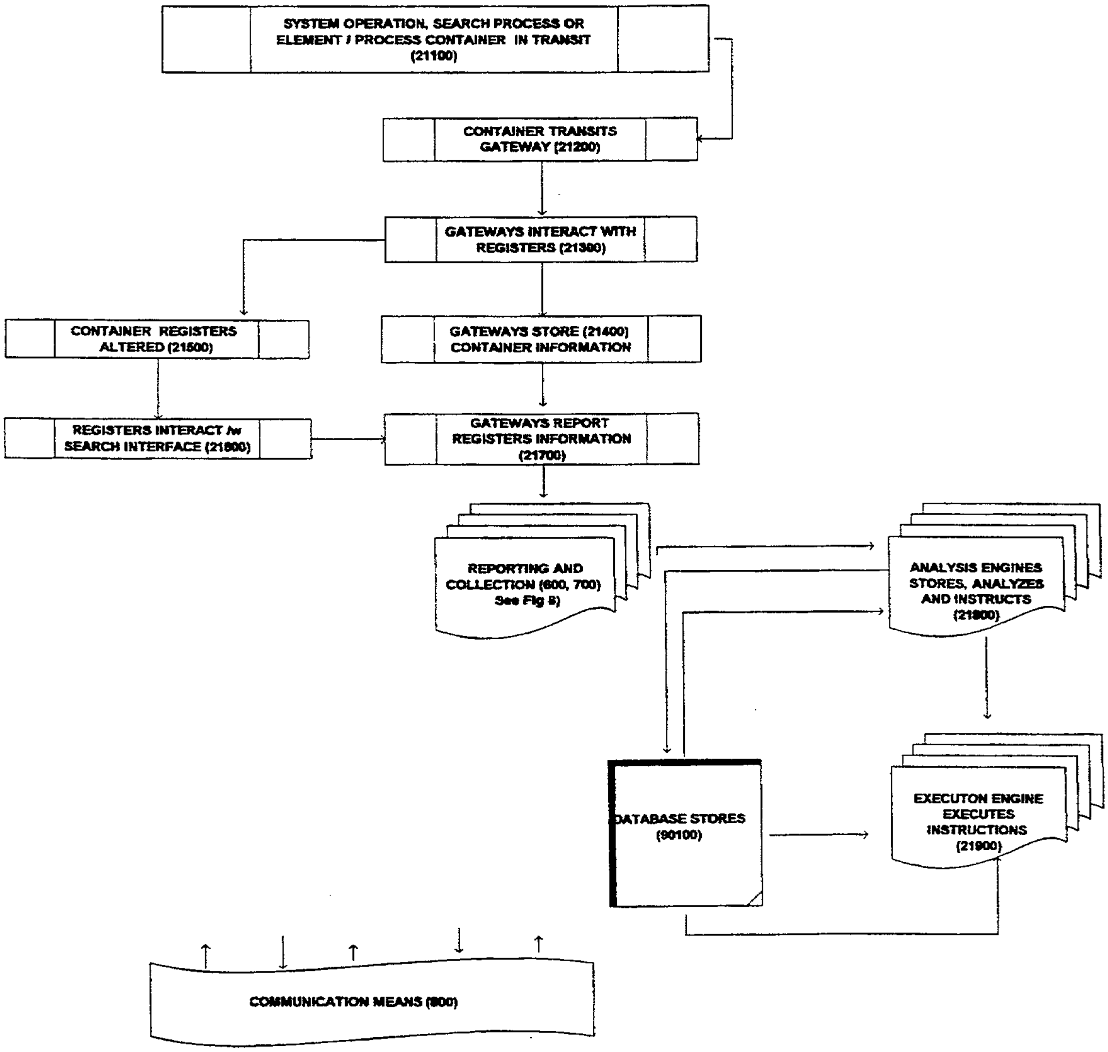

SYSTEM OPERATION, SEARCH PROCESS OR ELEMENT / PROCESS CONTAINER IN TRANSIT (21100)
CONTAINER TRANSITS GATEWAY (21200)
GATEWAYS INTERACT WITH REGISTERS (21300)
CONTAINER REGISTERS ALTERED (21500)
GATEWAYS STORE (21400) CONTAINER INFORMATION
REGISTERS INTERACT /w SEARCH INTERFACE (21600)
GATEWAYS REPORT REGISTERS INFORMATION (21700)
REPORTING AND COLLECTION (600, 700) See Fig 5)
ANALYSIS ENGINES STORES, ANALYZES AND INSTRUCTS (21800)
DATABASE STORES (90100)
EXECUTON ENGINE EXECUTES INSTRUCTIONS (21900)
COMMUNICATION MEANS (800)

FIG. 12

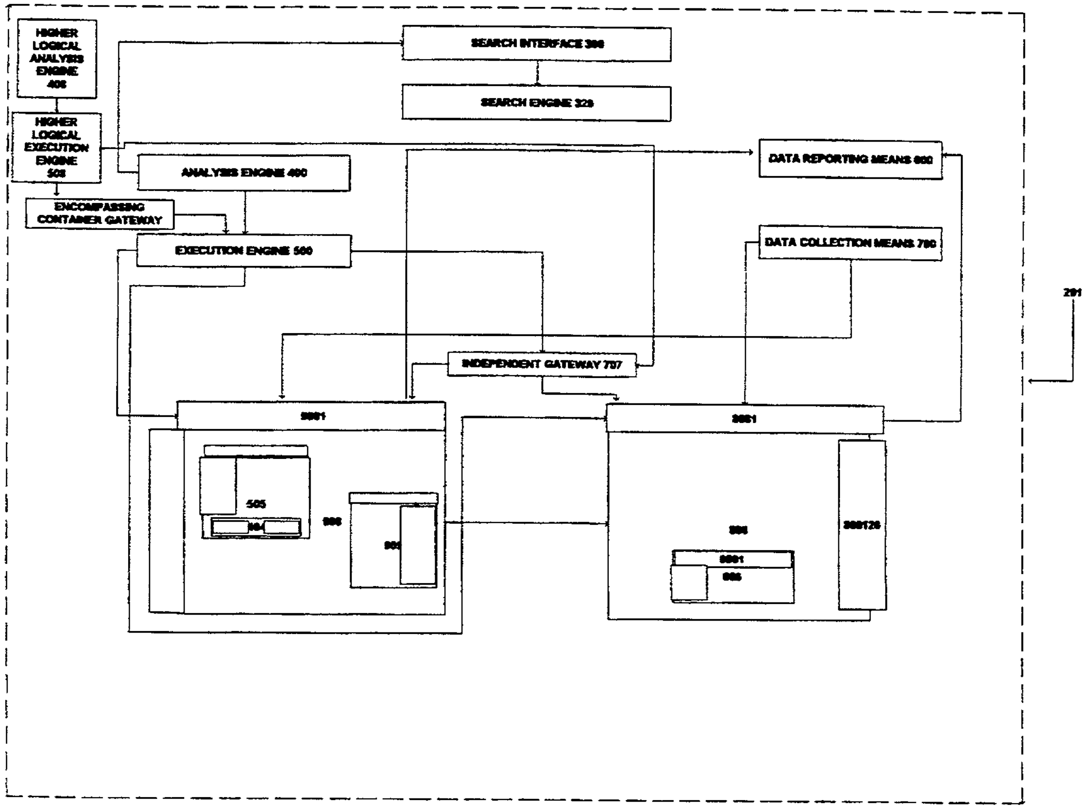

HIGHER LOGICAL ANALYSIS ENGINE 408
SEARCH INTERFACE 300
SEARCH ENGINE 320
HIGHER LOGICAL EXECUTION ENGINE 508
ANALYSIS ENGINE 400
DATA REPORTING MEANS 600
ENCOMPASSING CONTAINER GATEWAY
EXECUTION ENGINE 500
DATA COLLECTION MEANS 700
291
INDEPENDENT GATEWAY 707
9001
9001
505
504
906
901
906
900120
9001
906

FIG. 13 A

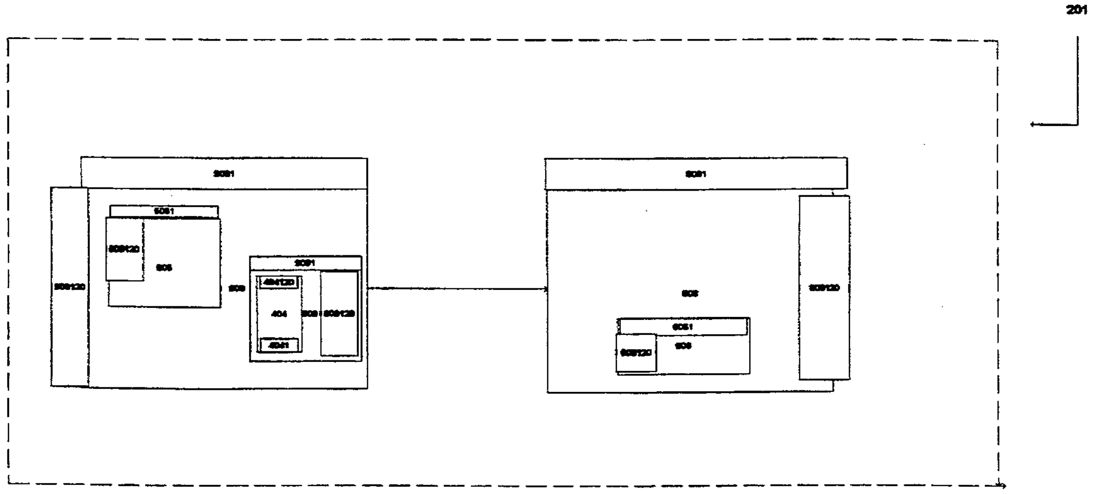

201
8081
808120
6061
606120
606
808
8081
404120
404
4041
808
808120
8081
808
808120
6061
606
606120

FIG. 13 B

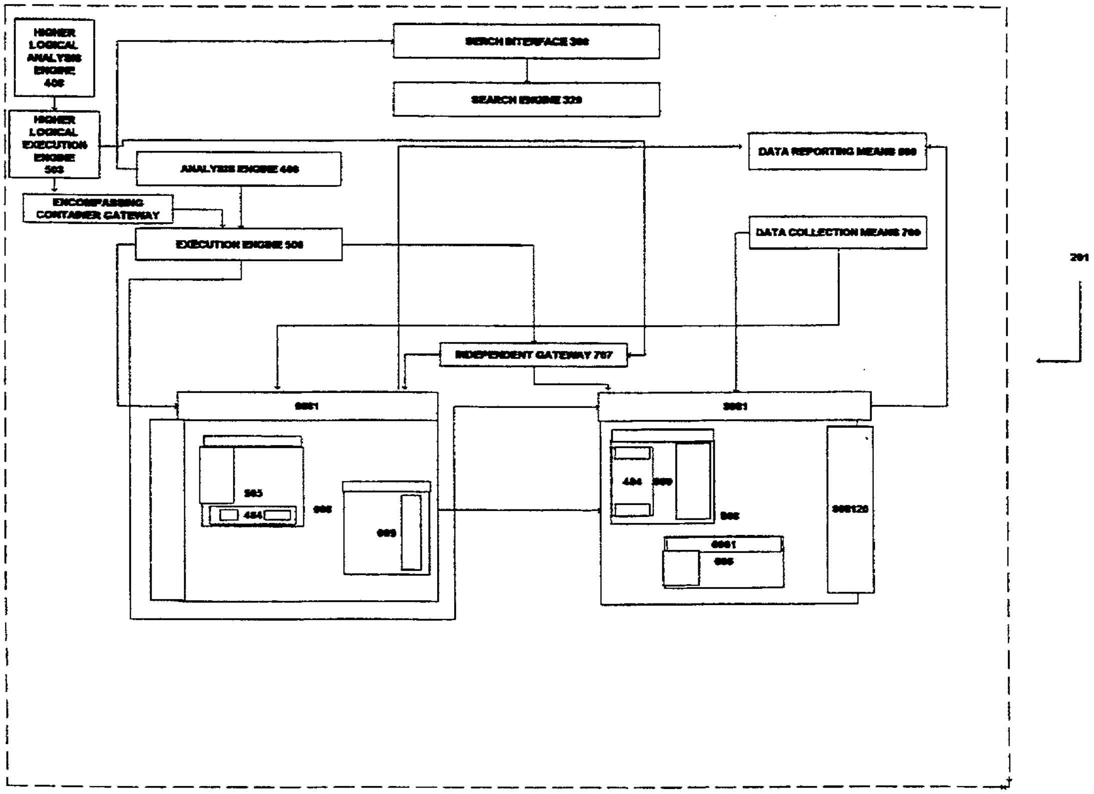

HIGHER LOGICAL ANALYSIS ENGINE 408
SERCH INTERFACE 300
SEARCH ENGINE 320
HIGHER LOGICAL EXECUTION ENGINE 503
ANALYSIS ENGINE 400
DATA REPORTING MEANS 800
ENCOMPASSING CONTAINER GATEWAY
EXECUTION ENGINE 500
DATA COLLECTION MEANS 700
201
INDEPENDENT GATEWAY 707
9001
3001
505
484
908
903
404
909
908
909120
6001
906

FIG. 13 C

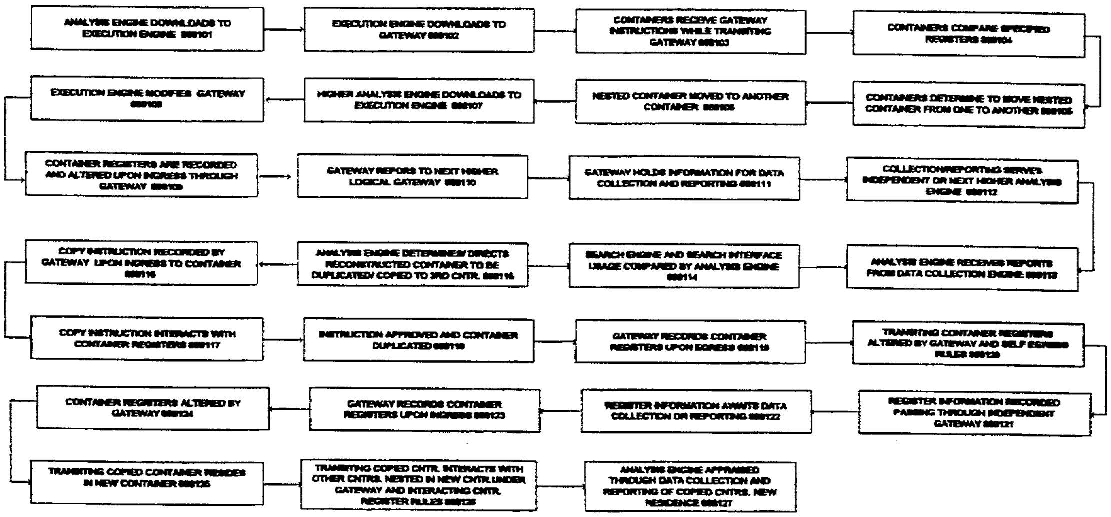

ANALYSIS ENGINE DOWNLOADS TO EXECUTION ENGINE 888101
EXECUTION ENGINE DOWNLOADS TO GATEWAY 888102
CONTAINERS RECEIVE GATEWAY INSTRUCTIONS WHILE TRANSITING GATEWAY 888103
CONTAINERS COMPARE SPECIFIED REGISTERS 888104
CONTAINERS DETERMINE TO MOVE NESTED CONTAINER FROM ONE TO ANOTHER 888105
NESTED CONTAINER MOVED TO ANOTHER CONTAINER 888106
HIGHER ANALYSIS ENGINE DOWNLOADS TO EXECUTION ENGINE 888107
EXECUTION ENGINE MODIFIES GATEWAY 888108
CONTAINER REGISTERS ARE RECORDED AND ALTERED UPON INGRESS THROUGH GATEWAY 888109
GATEWAY REPORS TO NEXT HIGHER LOGICAL GATEWAY 888110
GATEWAY HOLDS INFORMATION FOR DATA COLLECTION AND REPORTING 888111
COLLECTION/REPORTING SERVES INDEPENDENT OR NEXT HIGHER ANALYSIS ENGINE 888112
ANALYSIS ENGINE RECEIVES REPORTS FROM DATA COLLECTION ENGINE 888113
SEARCH ENGINE AND SEARCH INTERFACE USAGE COMPARED BY ANALYSIS ENGINE 888114
ANALYSIS ENGINE DETERMINES/ DIRECTS RECONSTRUCTED CONTAINER TO BE DUPLICATED/ COPIED TO 3RD CNTR. 888116
COPY INSTRUCTION RECORDED BY GATEWAY UPON INGRESS TO CONTAINER 888115
COPY INSTRUCTION INTERACTS WITH CONTAINER REGISTERS 888117
INSTRUCTION APPROVED AND CONTAINER DUPLICATED 888118
GATEWAY RECORDS CONTAINER REGISTERS UPON EGRESS 888119
TRANSITING CONTAINER REGISTERS ALTERED BY GATEWAY AND SELF EGRESS RULES 888120
REGISTER INFORMATION RECORDED PASSING THROUGH INDEPENDENT GATEWAY 888121
REGISTER INFORMATION AWAITS DATA COLLECTION OR REPORTING 888122
GATEWAY RECORDS CONTAINER REGISTERS UPON INGRESS 888123
CONTAINER REGISTERS ALTERED BY GATEWAY 888124
TRANSITING COPIED CONTAINER RESIDES IN NEW CONTAINER 888125
TRANSITING COPIED CNTR. INTERACTS WITH OTHER CNTRS. NESTED IN NEW CNTR. UNDER GATEWAY AND INTERACTING CNTR. REGISTER RULES 888126
ANALYSIS ENGINE APPRAISED THROUGH DATA COLLECTION AND REPORTING OF COPIED CNTRS. NEW RESIDENCE 888127

FIG. 13 D

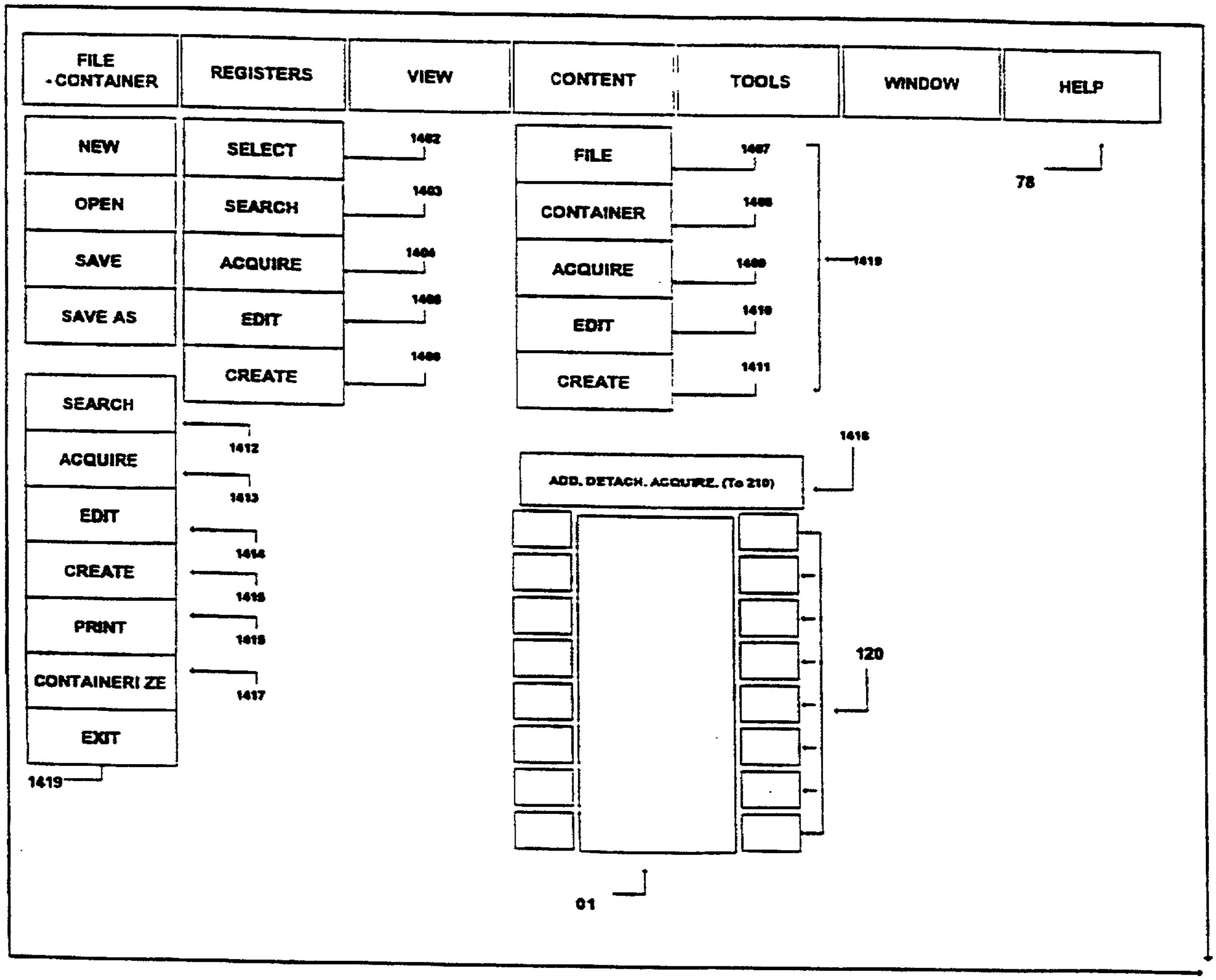

FILE -CONTAINER
REGISTERS
VIEW
CONTENT
TOOLS
WINDOW
HELP
NEW
OPEN
SAVE
SAVE AS
SELECT
SEARCH
ACQUIRE
EDIT
CREATE
1402
1403
1404
1405
1406
FILE
CONTAINER
ACQUIRE
EDIT
CREATE
1407
1408
1409
1410
1411
1419
78
SEARCH
ACQUIRE
EDIT
CREATE
PRINT
CONTAINERI ZE
EXIT
1412
1413
1414
1415
1416
1417
1419
1418
ADD. DETACH. ACQUIRE. (To 210)
120
01

FIG. 14

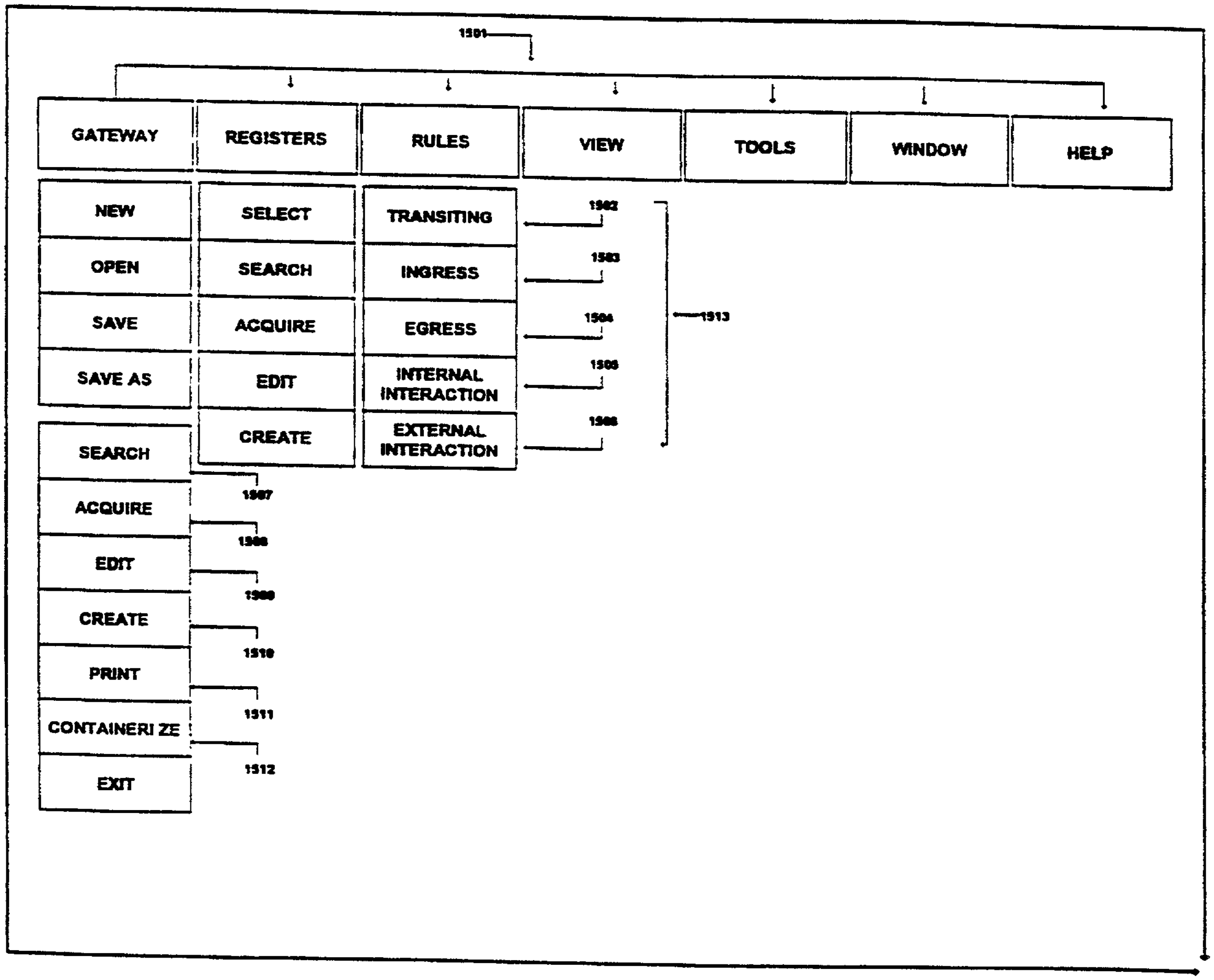

1501
GATEWAY
REGISTERS
RULES
VIEW
TOOLS
WINDOW
HELP
NEW
OPEN
SAVE
SAVE AS
SEARCH
ACQUIRE
EDIT
CREATE
PRINT
CONTAINERI ZE
EXIT
SELECT
SEARCH
ACQUIRE
EDIT
CREATE
TRANSITING
INGRESS
EGRESS
INTERNAL INTERACTION
EXTERNAL INTERACTION
1502
1503
1504
1505
1506
1513
1507
1508
1509
1510
1511
1512

FIG. 15

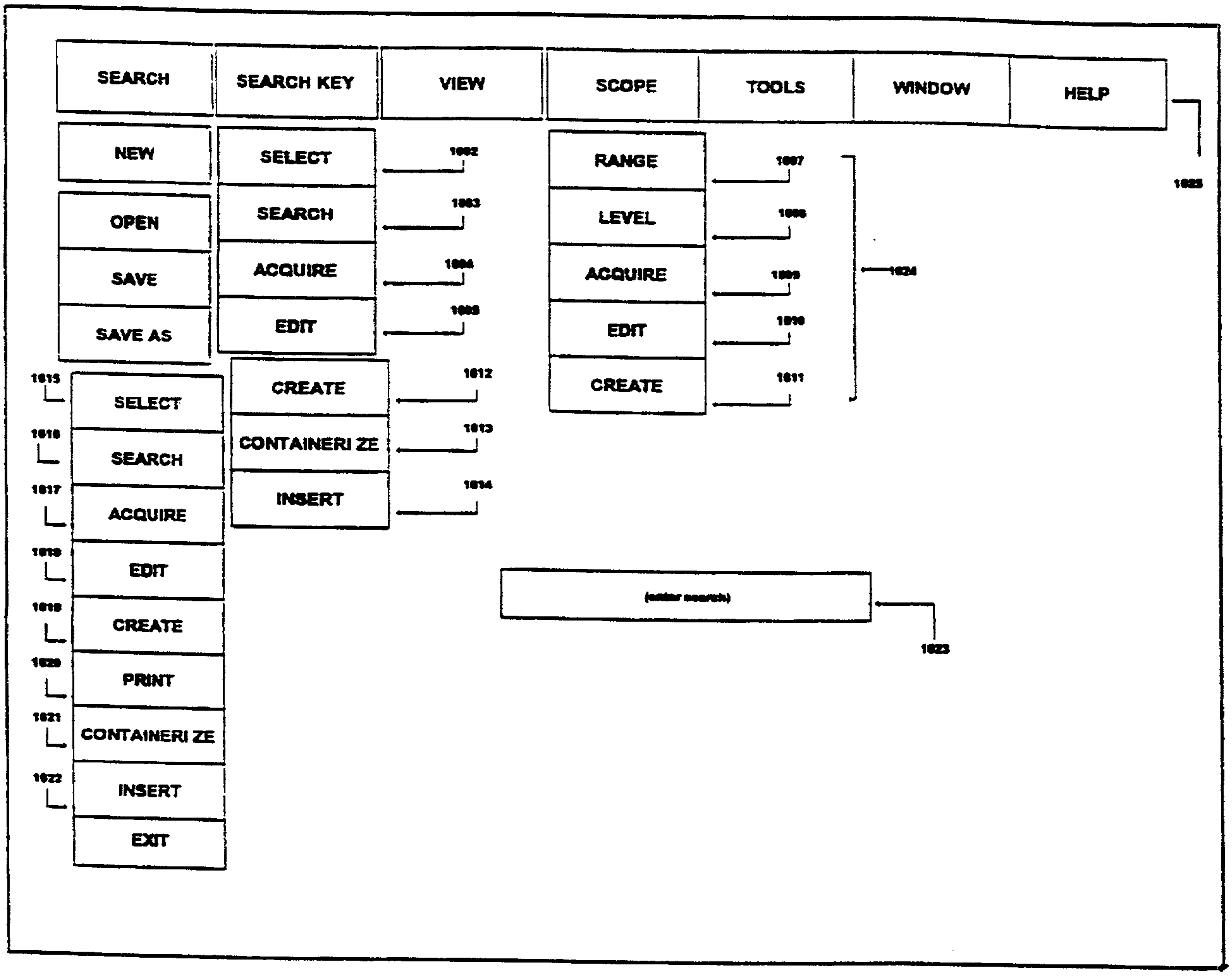

SEARCH
SEARCH KEY
VIEW
SCOPE
TOOLS
WINDOW
HELP
1625
NEW
OPEN
SAVE
SAVE AS
SELECT
1602
SEARCH
1603
ACQUIRE
1604
EDIT
1605
RANGE
1607
LEVEL
1608
ACQUIRE
1609
EDIT
1610
CREATE
1611
1624
CREATE
1612
CONTAINERI ZE
1613
INSERT
1614
1615
SELECT
1616
SEARCH
1617
ACQUIRE
1618
EDIT
1619
CREATE
1620
PRINT
1621
CONTAINERI ZE
1622
INSERT
EXIT
(enter search)
1623

FIG. 16

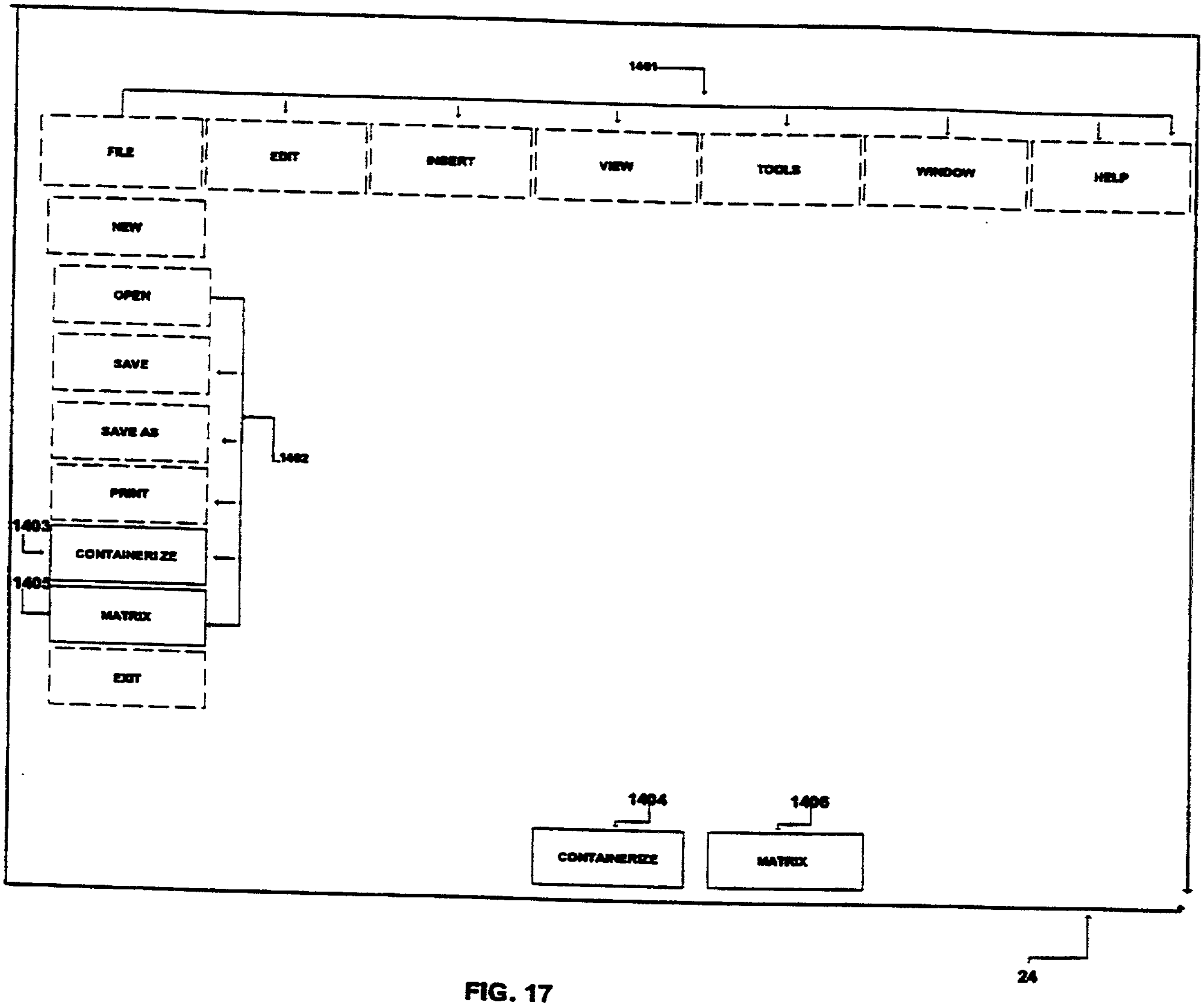

1401
FILE
EDIT
INSERT
VIEW
TOOLS
WINDOW
HELP
NEW
OPEN
SAVE
SAVE AS
1402
PRINT
1403
CONTAINERIZE
1405
MATRIX
EXIT
1404
CONTAINERIZE
1406
MATRIX
24

FIG. 17

PTO/SB/01 (6-95) (modified)
Approved for use through 10/31/96 OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

0010/PTO
Rev. 6/95

U.S. Department of Commerce
Patent and Trademark Office

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION

[X] Declaration Submitted with Initial Filing OR [ ] Declaration Submitted after Initial Filing

Attorney Docket Number	**3726**
First Named Inventor	**Michael De Angelo**
COMPLETE IF KNOWN	
Application Number	**Unknown**
Filing Date	**Even Date Herewith**
Group Art Unit	**Unknown**
Examiner Name	**Unknown**

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

the specification of which (*Title of the Invention*)

[ ] is attached hereto

OR

[X] was filed on (MM/DD/YYYY) **[01/28/1999]** as United States Application Number or PCT International Application Number **[PCT/US99/01988]** and was amended on (MM/DD/YYYY) [________________] (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above. .

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37 Code of Federal Regulations. § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code § 119 (a)-(d) or § 385(b) of any foreign application(s) for patent or inventor's certificate, or § 365 (a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]

[ ] Additional foreign application numbers are listed on a supplemental priority sheet attached hereto:

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	[ ] Additional provisional application numbers are listed on a supplemental sheet attached hereto.
60/073,209	**01/30/1998**	

DECLARATION	**Page 2**

I hereby claim the benefit under Title 35, United States Code § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (*if applicable*)
	PCT/US99/01988	**01/28/1999**	

[ ] Additional U.S. or PCT international application numbers are listed on a supplemental priority sheet attached hereto.

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Name	Registration Number	Name	Registration Number
Greg T. Sueoka	33,800		
James K. Okamoto	40,110		

[ ] Additional attorney(s) and/or agent(s) named on a supplemental sheet attached hereto.

Please direct all correspondence to:

Greg T. Sueoka
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
U.S.A.

Telephone	**(650) 858-7194**	Fax	**(650) 494-1417**

②

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name of Sole or First Inventor: [ ] A petition has been filed for this unsigned inventor

Given Name	**Michael**	Middle Initial		Family Name	**De Angelo**	Suffix e.g. Jr.	
Inventor's Signature	Michael DeAngelo			Date	April 5, 1999		
Residence: City	**Santa Barbara**	State	**CA** CA	Country	**USA**	Citizenship	
Mailing Address	**1324 J State Street, Suite 290**						
Mailing Address							
City	**Santa Barbara**	State	**CA**	Zip	**93101**	Country	**USA**

[ ] Additional inventors are being named on supplemental sheet(s) attached hereto

1-00

09/[illegible]160

ISSUE CLASSIFICATION	
Class	Subclass

[illegible]

PATENT NUMBER

U.S. UTILITY PATENT APPLICATION

O.I.P.E.	PATENT DATE
123 SCANNED BK Q.A. RG	

SECTOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
	707	1	2171	Nguyen, Cam Linh. ~~Metjahic~~, S.

FILED WITH: □ DISK (CRF) □ FICHE
(Attached in pocket on right inside flap)

Best Available Copy

[redacted]

PREPARED AND APPROVED FOR ISSUE

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)						
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					
INTERNATIONAL CLASSIFICATION								
			□ Continued on Issue Slip Inside File Jacket					

□ TERMINAL DISCLAIMER	DRAWINGS			CLAIMS ALLOWED	
	Sheets Drwg.	Figs. Drwg.	Print Fig.	Total Claims	Print Claim for O.G.

□ a) The term of this patent subsequent to ____________ (date) has been disclaimed.

□ b) The term of this patent shall not extend beyond the expiration date of U.S Patent. No. ____________

□ c) The terminal ____months of this patent have been disclaimed.

____________ (Assistant Examiner) ______ (Date)

____________ (Primary Examiner) ______ (Date)

____________ (Legal Instruments Examiner) ______ (Date)

NOTICE OF ALLOWANCE MAILED

ISSUE FEE	
Amount Due	Date Paid

ISSUE BATCH NUMBER

WARNING:
The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 6/98)

(LABEL AREA)

Best Available Copy

SEARCHED

Class	Sub.	Date	Exmr.
707	103 10	1/29/03	Linh

SEARCH NOTES
(INCLUDING SEARCH STRATEGY)

	Date	Exmr.
WEST Search consulting Amsbury Wayne (707)	1/29/03	Linh

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.

(RIGHT OUTSIDE)

ISSUE SLIP STAPLE AREA (for additional cross references)

POSITION	INITIALS	ID NO.	DATE
FEE DETERMINATION			
O.I.P.E. CLASSIFIER		6	3-1-00
FORMALITY REVIEW			

INDEX OF CLAIMS

✓ .... Rejected
= .... Allowed
— (Through numeral)... Canceled
÷ .... Restricted
N .... Non-elected
I .... Interference
A .... Appeal
O .... Objected

3

Claim		Date									
Final	Original	1/30/03									
	1	✓									
	2										
	3										
	4										
	5										
	6										
	7										
	~~8~~										
	9										
	10										
	~~11~~										
	~~12~~										
	~~13~~										
	14										
	15										
	16										
	17										
	18										
	19										
	20										
	21										
	22										
	23										
	24										
	25										
	26										
	27										
	28										
	29										
	30										
	31										
	32										
	33										
	34										
	35										
	36										
	37										
	38										
	39										
	40										
	41										
	42										
	43										
	44										
	45										
	46										
	47										
	48										
	49										
	50										

Claim		Date									
Final	Original										
	51										
	52										
	53										
	54										
	55										
	56										
	57										
	58										
	59										
	60										
	61										
	62										
	63										
	64										
	65										
	66										
	67										
	68										
	69										
	70										
	71										
	72										
	73										
	74										
	75										
	76										
	77										
	78										
	79										
	80										
	81										
	82										
	83										
	84										
	85										
	86										
	87										
	88										
	89										
	90										
	91										
	92										
	93										
	94										
	95										
	96										
	97										
	98										
	99										
	100										

Claim		Date									
Final	Original										
	101										
	102										
	103										
	104										
	105										
	106										
	107										
	108										
	109										
	110										
	111										
	112										
	113										
	114										
	115										
	116										
	117										
	118										
	119										
	120										
	121										
	122										
	123										
	124										
	125										
	126										
	127										
	128										
	129										
	130										
	131										
	132										
	133										
	134										
	135										
	136										
	137										
	138										
	139										
	140										
	141										
	142										
	143										
	144										
	145										
	146										
	147										
	148										
	149										
	150										

If more than 150 claims or 10 actions
staple additional sheet here

(LEFT INSIDE)

FORM PTO-1390 (REV. 1-98) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY'S DOCKET NUMBER 3726 US
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371	U.S. APPLICATION NO. (If known, see 37 CFR 1.5) 09/284113 Not Yet Known

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/US99/01988	28 January 1999	30 January 1998

TITLE OF INVENTION
System And Method For Creating And Manipulating Information Containers With Dynamic Registers

APPLICANT(S) FOR DO/EO/US
Michael De Angelo

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☑ This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.
2. □ This is a SECOND or SUBSEQUENT submission of items concerning a filing under 35 U.S.C. 371.
3. ☑ This express request to begin national examination procedures (35 U.S.C. 371 (f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. □ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☑ A copy of the International Application as filed (35 U.S.C. 371(c)(2)).
   a. □ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. □ has been transmitted by the International Bureau.
   c. ☑ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. □ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☑ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3)).
   a. □ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. □ have been transmitted by the International Bureau.
   c. ☑ have not been made; however, the time limit for making such amendments has NOT expired.
   d. □ have not been made and will not be made.
8. □ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☑ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. □ A translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern document(s) or information included:

11. □ An Information Disclosure Statement under 37 CFR 1.97 and 1.9
12. ☑ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. □ A FIRST preliminary amendment.
    □ A SECOND or SUBSEQUENT preliminary amendment.
14. □ A substitute specification.
15. □ A change of power of attorney and/or address letter.
16. ☑ Other items or information: A Verified Statement Claiming Small Entity Status

U.S. APPLICATION NO. (if known, CFR 1.5) Not Yet Known	INTERNATIONAL APPLICATION NO. PCT/US99/01988	ATTORNEY'S DOCKET NUMBER 3726 US

17. ☑ The following fees are submitted: BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(5)): Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO.........$970.00 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO..$840.00 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO........$760.00 International preliminary examination fee (37 CFR 1.482) paid to USPTO but all claims did not satisfy provisions of PCT Article 33(2)-(4)......$670.00 International preliminary examination fee (37 CFR 1.482) paid to USPTO and all claims satisfied provisions of PCT Article 33(2)-(4).............$96.00				CALCULATIONS	PTO USE ONLY
ENTER APPROPRIATE BASIC FEE AMOUNT =				$760.00	
Surcharge of $130.00 for furnishing the oath or declaration later than □ 20 □ 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				$0	
CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total claims	36 - 20 =	16	x $18.00	$288.00	
Independent claims	3 - 3 =	0	x $78,00	$0	
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+ $260.00	$0	
TOTAL OF ABOVE CALCULATIONS =				$1048.00	
Reduction of 1/2 for filing by small entity, if applicable. A Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28). +				$524.00	
SUBTOTAL =				$524.00	
Processing fee of $130.00 for furnishing the English translation later than □ 20 □ 30 months from the earliest claimed priority date (37 CFR 1.492(f)).				$0	
TOTAL NATIONAL FEE =				$524.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). $40.00 per property +				$40.00	
TOTAL FEES ENCLOSED =				$564.00	
				Amount to be rendered:	$564.00
				charged:	$

a. ☑ A check in the amount of $ 564.00 to cover the above fees is enclosed.

b. □ Please charge my Deposit Account No. ______________ in the amount of $ ____________ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☑ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 19-2555. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137 (a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

SIGNATURE

Greg T. Sueoka
FENWICK & WEST LLP

Greg T. Sueoka
NAME

Two Palo Alto Square
Palo Alto, CA 94306

33,800
REGISTRATION NUMBER

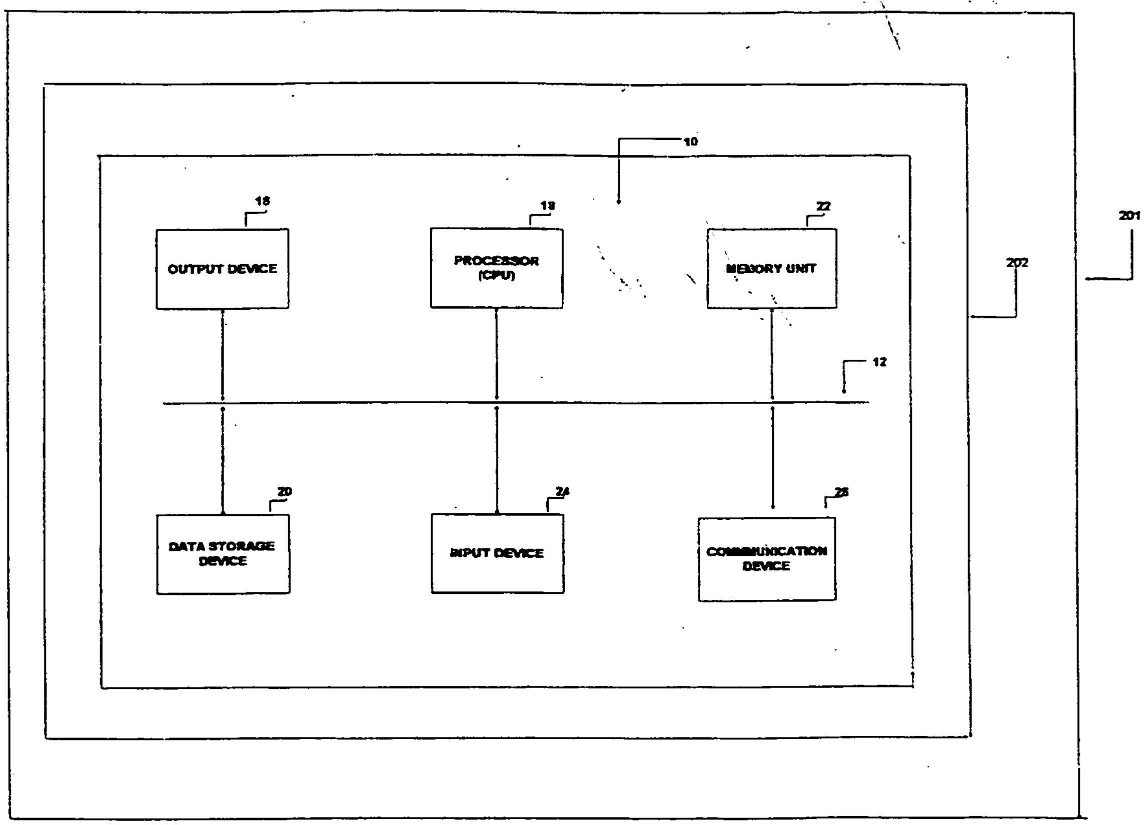

10
16
OUTPUT DEVICE
18
PROCESSOR (CPU)
22
MEMORY UNIT
12
20
DATA STORAGE DEVICE
24
INPUT DEVICE
26
COMMUNICATION DEVICE
202
201

FIG. 1

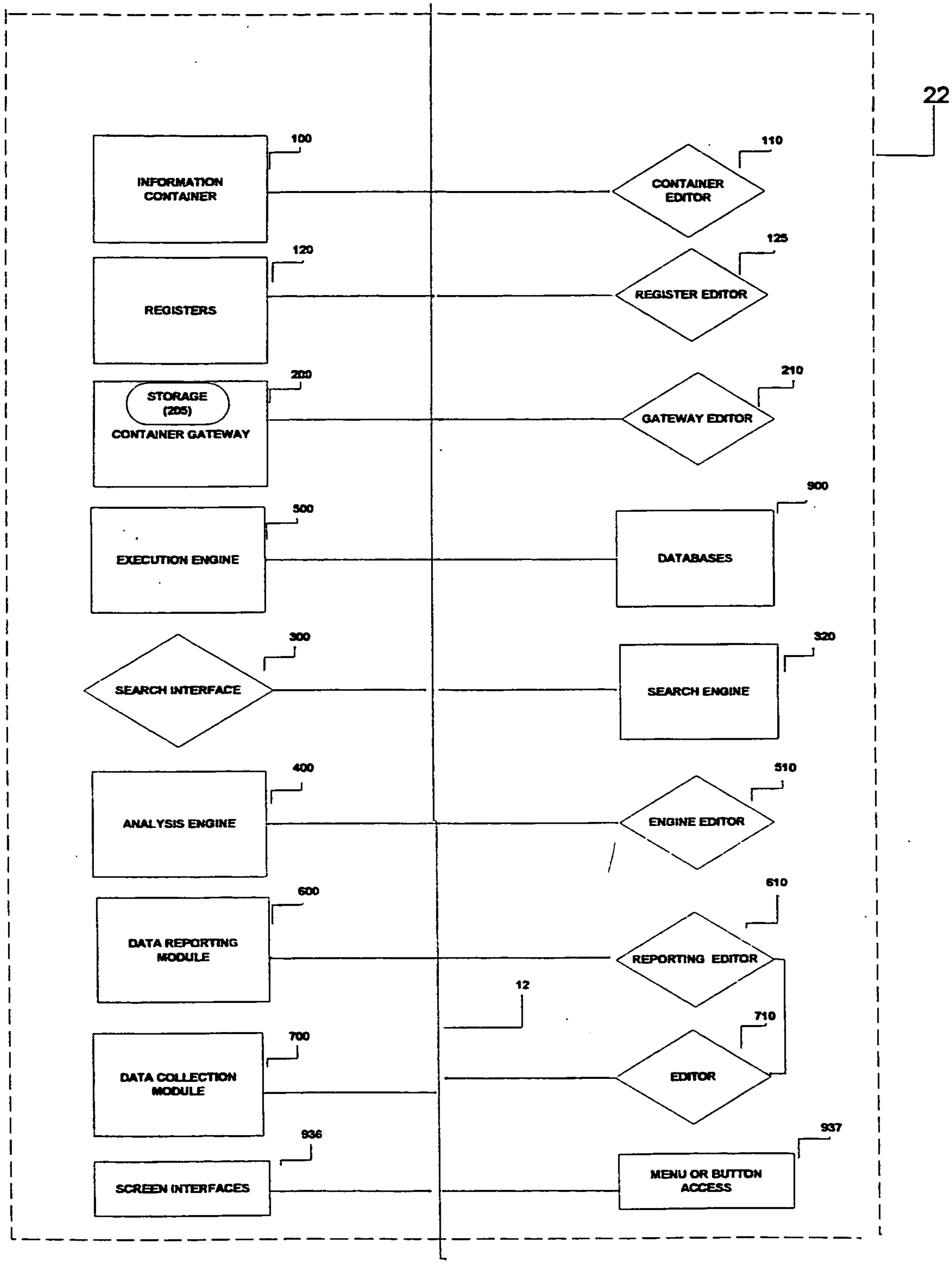

22
INFORMATION CONTAINER
100
CONTAINER EDITOR
110
REGISTERS
120
REGISTER EDITOR
125
STORAGE (205)
CONTAINER GATEWAY
200
GATEWAY EDITOR
210
EXECUTION ENGINE
500
DATABASES
900
SEARCH INTERFACE
300
SEARCH ENGINE
320
ANALYSIS ENGINE
400
ENGINE EDITOR
510
DATA REPORTING MODULE
600
REPORTING EDITOR
610
12
DATA COLLECTION MODULE
700
EDITOR
710
SCREEN INTERFACES
936
MENU OR BUTTON ACCESS
937

FIG. 2 A

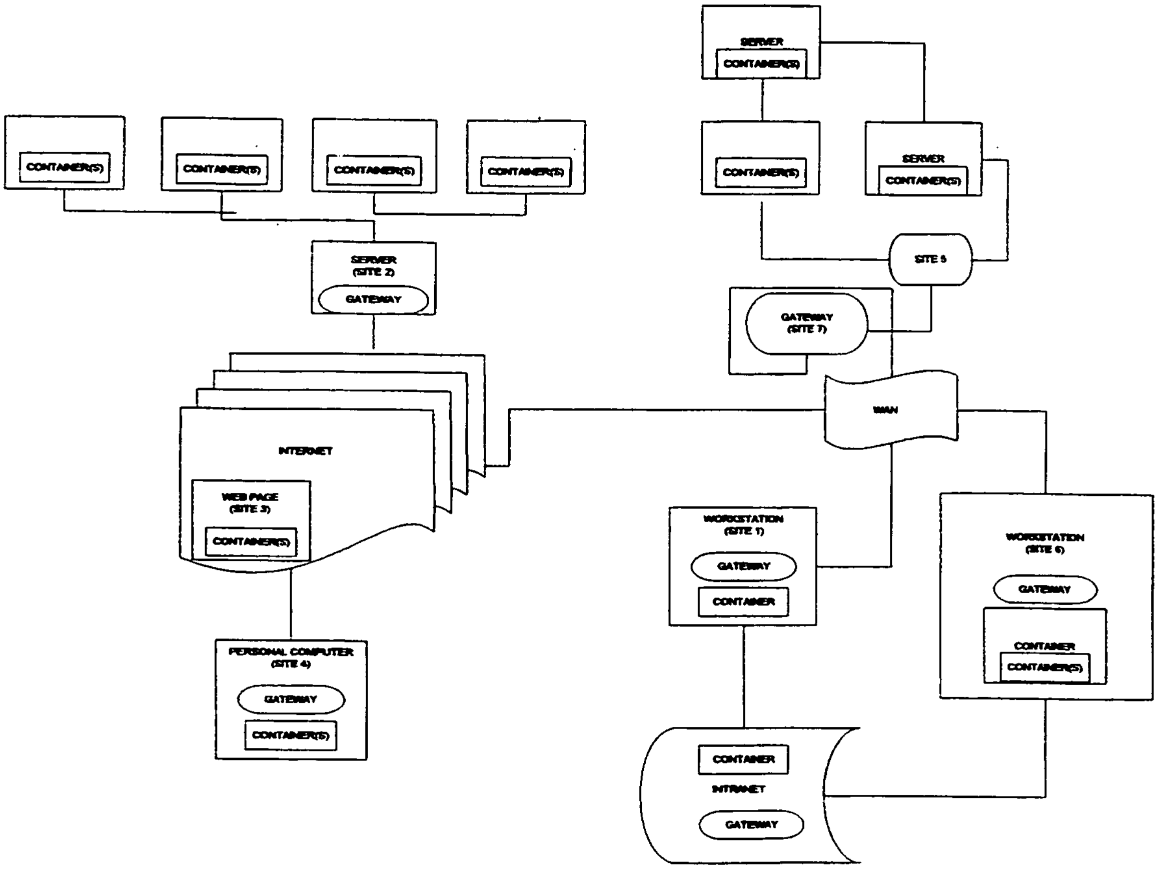

SERVER
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
CONTAINER(S)
SERVER
CONTAINER(S)
SERVER
(SITE 2)
GATEWAY
SITE 5
GATEWAY
(SITE 7)
WAN
INTERNET
WEB PAGE
(SITE 3)
CONTAINER(S)
WORKSTATION
(SITE 1)
GATEWAY
CONTAINER
WORKSTATION
(SITE 6)
GATEWAY
CONTAINER
CONTAINER(S)
PERSONAL COMPUTER
(SITE 4)
GATEWAY
CONTAINER(S)
CONTAINER
INTRANET
GATEWAY

FIG. 2 B

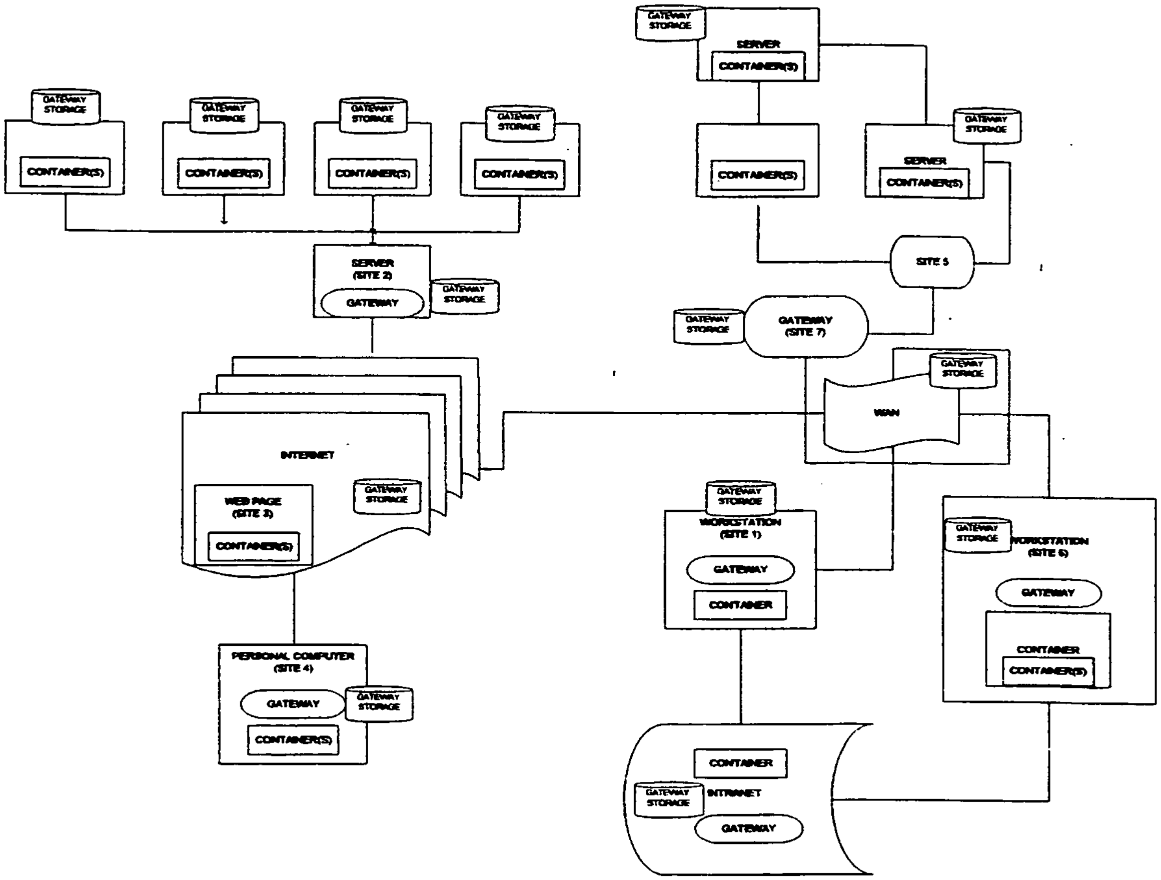

GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
CONTAINER(S)
SERVER (SITE 2)
GATEWAY
GATEWAY STORAGE
INTERNET
WEB PAGE (SITE 3)
CONTAINER(S)
GATEWAY STORAGE
PERSONAL COMPUTER (SITE 4)
GATEWAY
GATEWAY STORAGE
CONTAINER(S)
GATEWAY STORAGE
SERVER
CONTAINER(S)
CONTAINER(S)
GATEWAY STORAGE
SERVER
CONTAINER(S)
SITE 5
GATEWAY STORAGE
GATEWAY (SITE 7)
GATEWAY STORAGE
WAN
GATEWAY STORAGE
WORKSTATION (SITE 1)
GATEWAY
CONTAINER
GATEWAY STORAGE
WORKSTATION (SITE 6)
GATEWAY
CONTAINER
CONTAINER(S)
CONTAINER
GATEWAY STORAGE
INTRANET
GATEWAY

FIG. 2 B 1

5/30

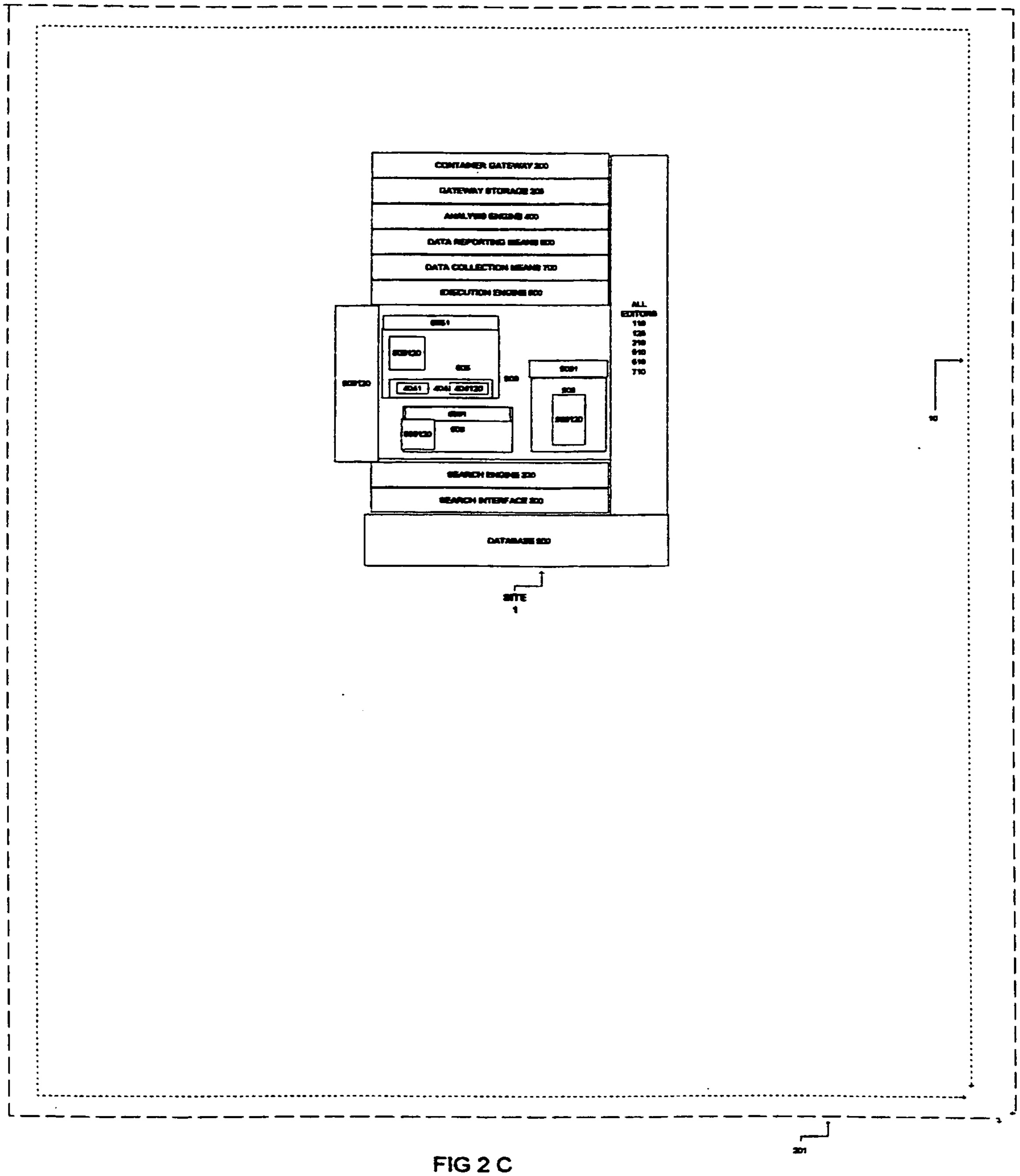

CONTAINER GATEWAY 200
GATEWAY STORAGE 205
ANALYSIS ENGINE 400
DATA REPORTING MEANS 600
DATA COLLECTION MEANS 700
EXECUTION ENGINE 800
ALL
EDITORS
110
120
210
910
610
710
809120
8051
809120
805
4041
404I
404120
808
8091
808
809120
8081
806
809120
SEARCH ENGINE 230
SEARCH INTERFACE 300
DATABASE 500
SITE
1
10
201

FIG 2 C

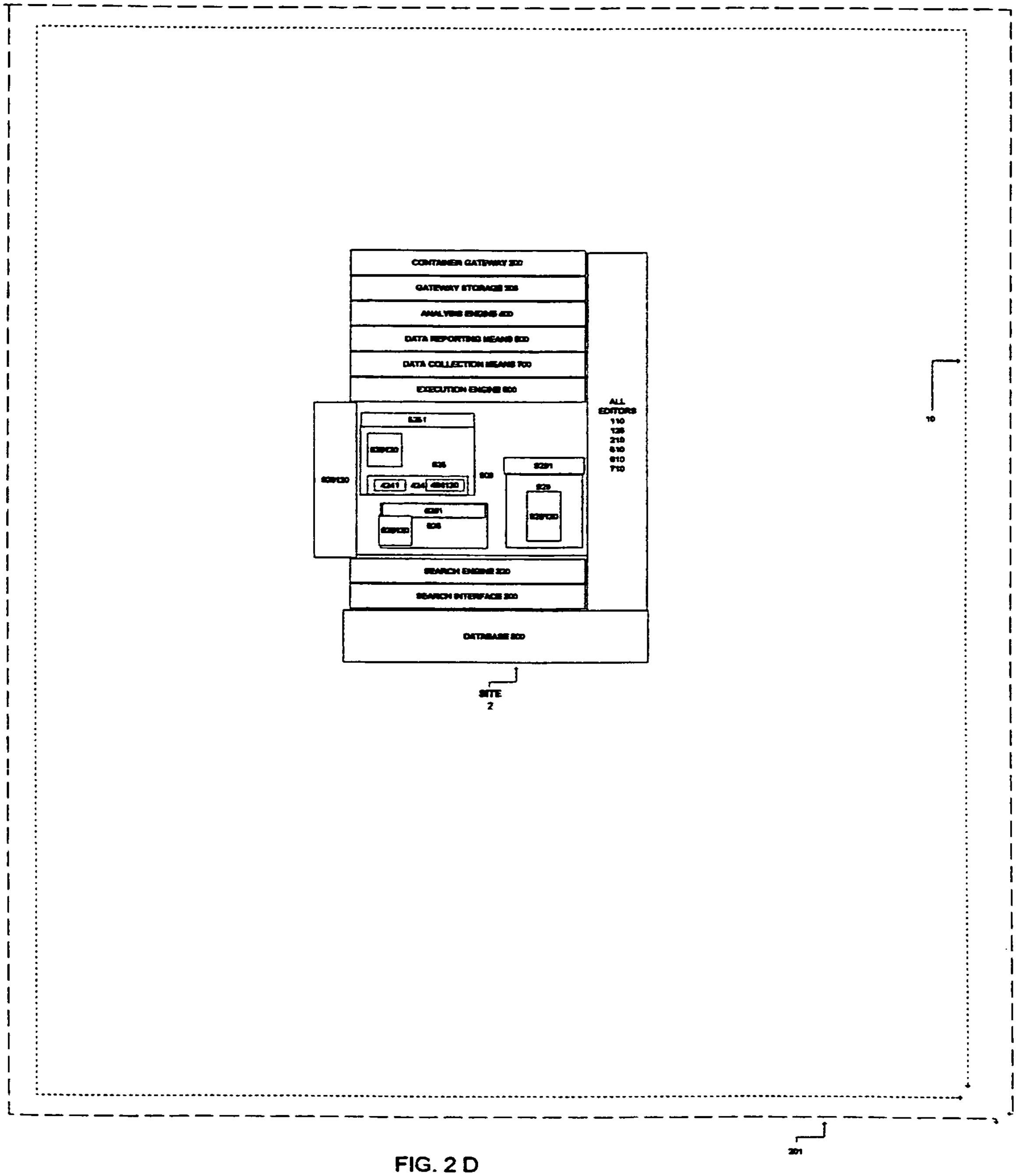

CONTAINER GATEWAY 300
GATEWAY STORAGE 305
ANALYSIS ENGINE 400
DATA REPORTING MEANS 500
DATA COLLECTION MEANS 700
EXECUTION ENGINE 600
ALL
EDITORS
110
125
210
510
610
710
8251
826/120
825
4241
424
4241/130
829/130
829
8291
829
829/120
8261
826/120
826
SEARCH ENGINE 220
SEARCH INTERFACE 200
DATABASE 800
SITE
2
10
201

FIG. 2 D

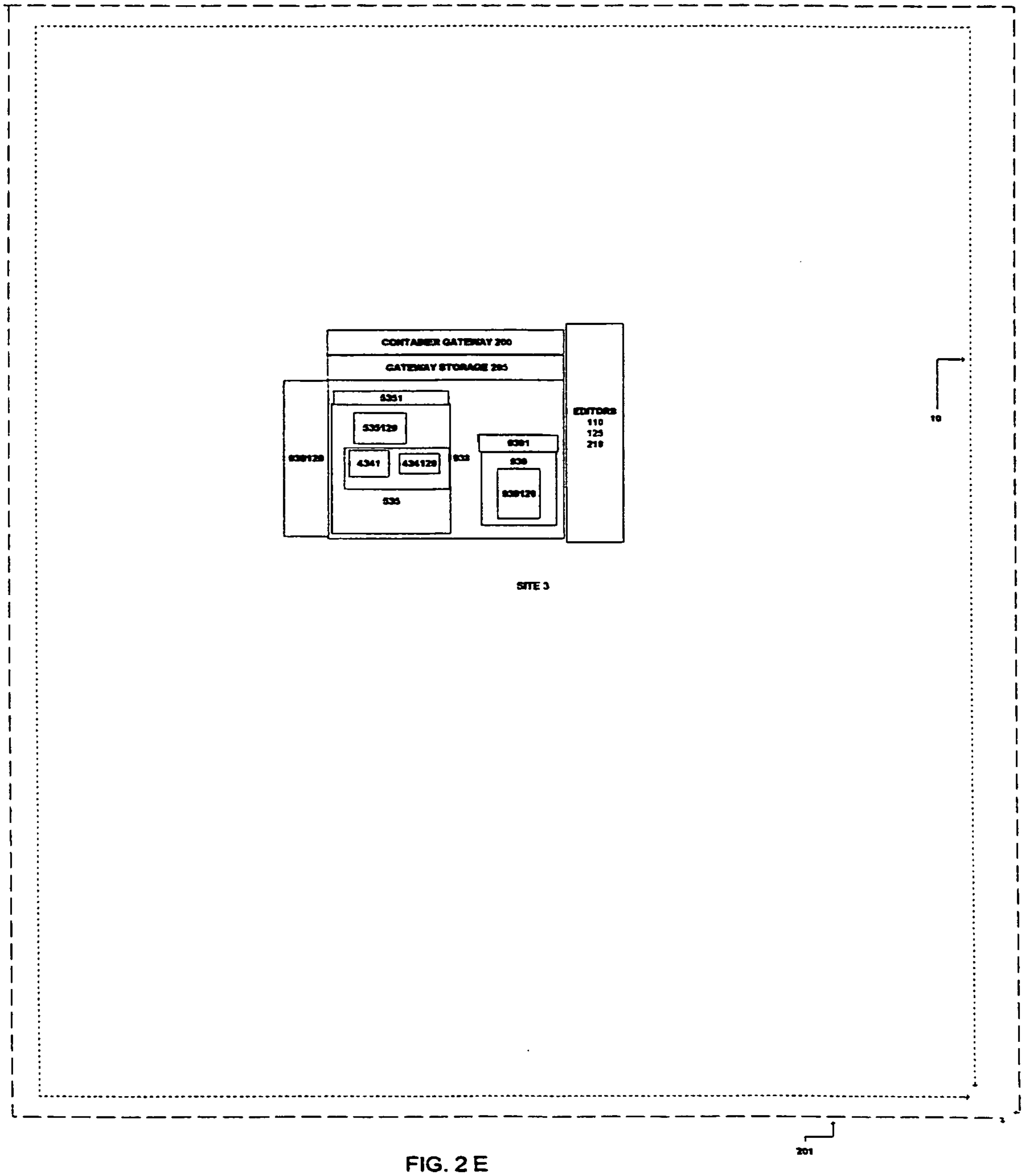

CONTAINER GATEWAY 200
GATEWAY STORAGE 205
530129
5351
535129
4341
434129
535
532
5301
530
530129
EDITORS
110
125
210
SITE 3
10
201

FIG. 2 E

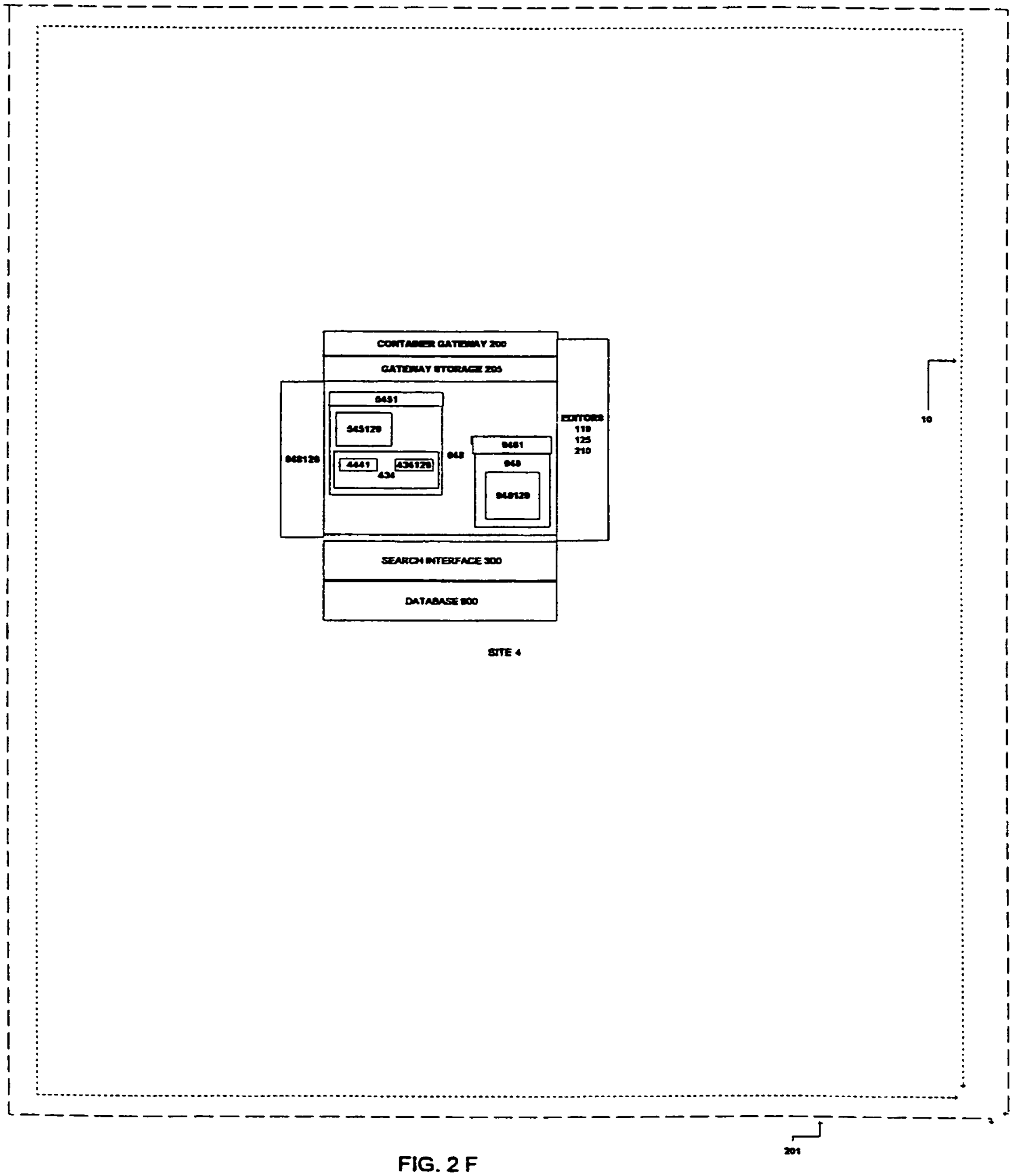

CONTAINER GATEWAY 200
GATEWAY STORAGE 205
8431
843129
4441
434129
434
842
848129
8481
848
848129
EDITORS
110
125
210
SEARCH INTERFACE 300
DATABASE 800
SITE 4
10
201

FIG. 2 F

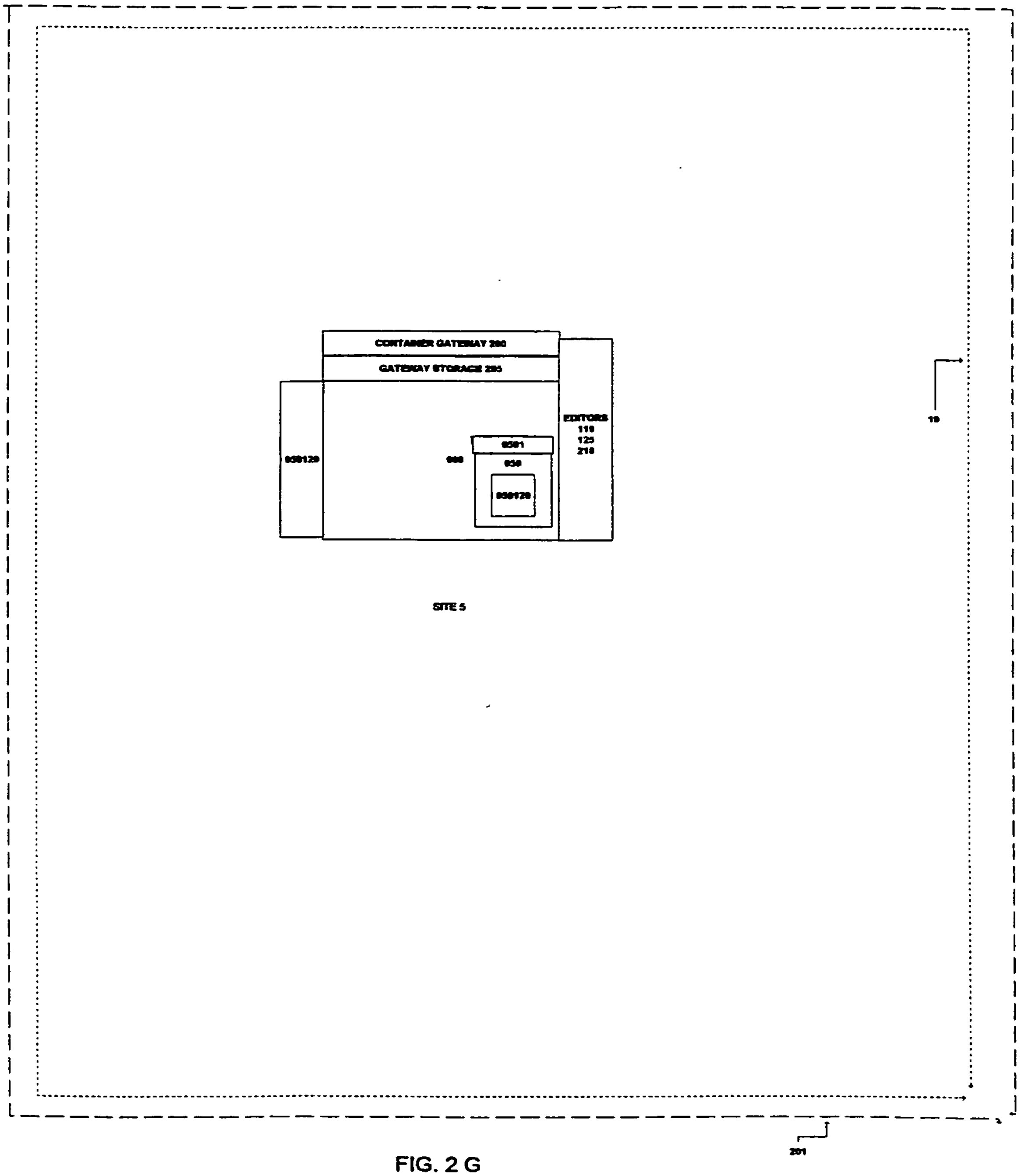

CONTAINER GATEWAY 280
GATEWAY STORAGE 285
950120
900
9501
950
950120
EDITORS
110
125
210
10
SITE 5
201

FIG. 2 G

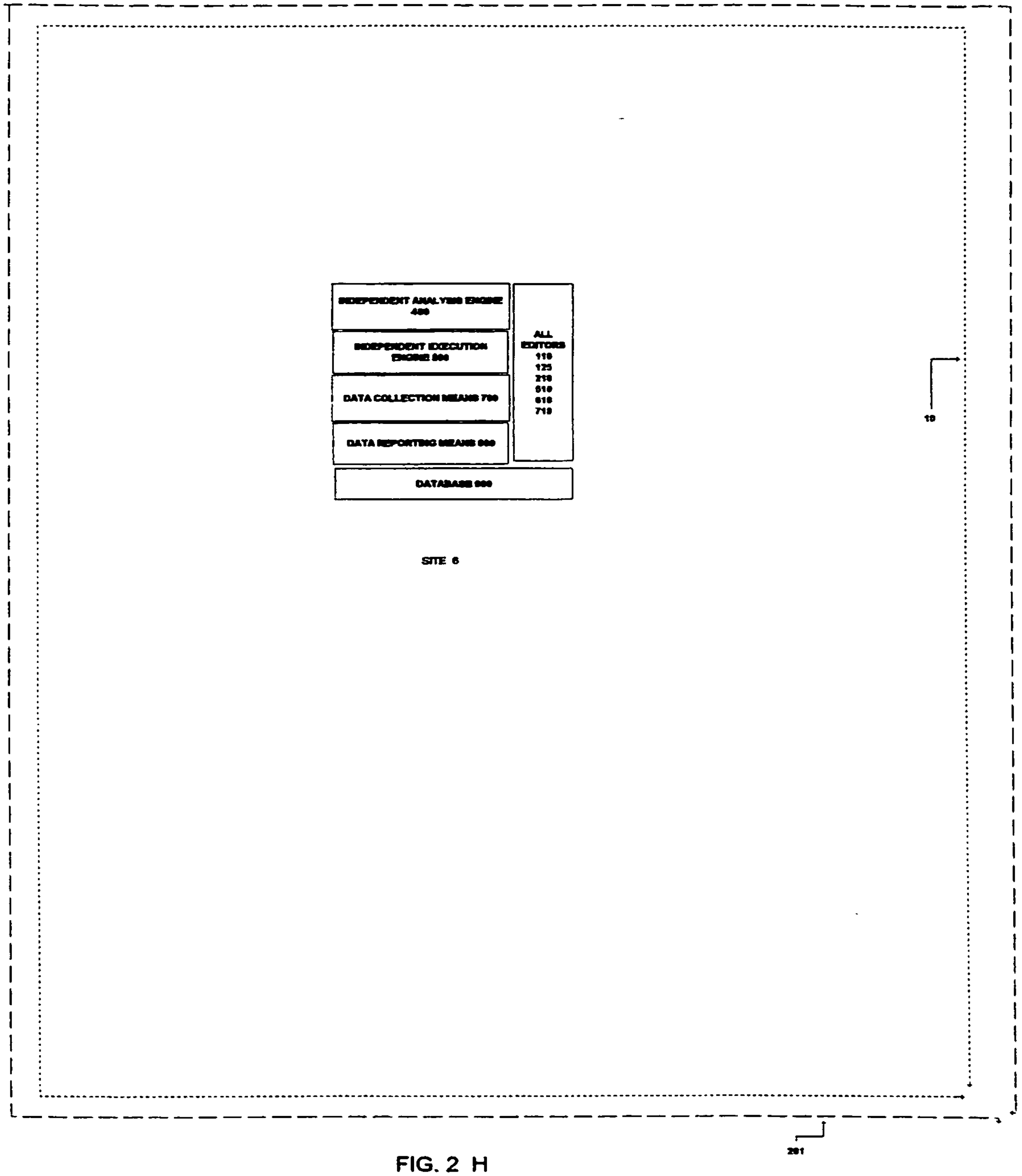

INDEPENDENT ANALYSIS ENGINE 400
INDEPENDENT EXECUTION ENGINE 500
DATA COLLECTION MEANS 700
DATA REPORTING MEANS 800
ALL EDITORS 110 125 210 510 610 710
DATABASE 900
SITE 6
10
201

FIG. 2 H

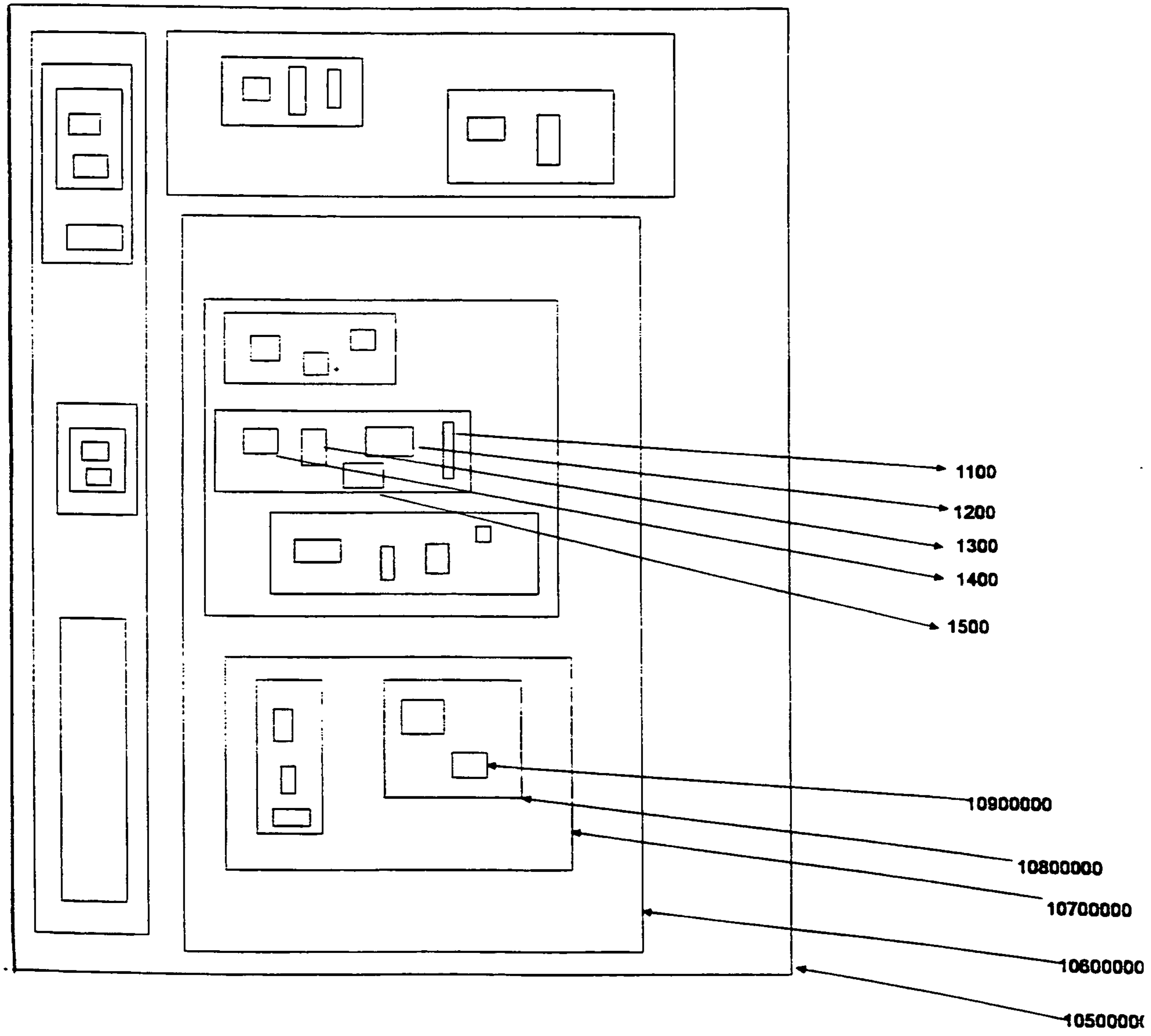

1100
1200
1300
1400
1500
10900000
10800000
10700000
10600000
1050000(

FIG. 3 A

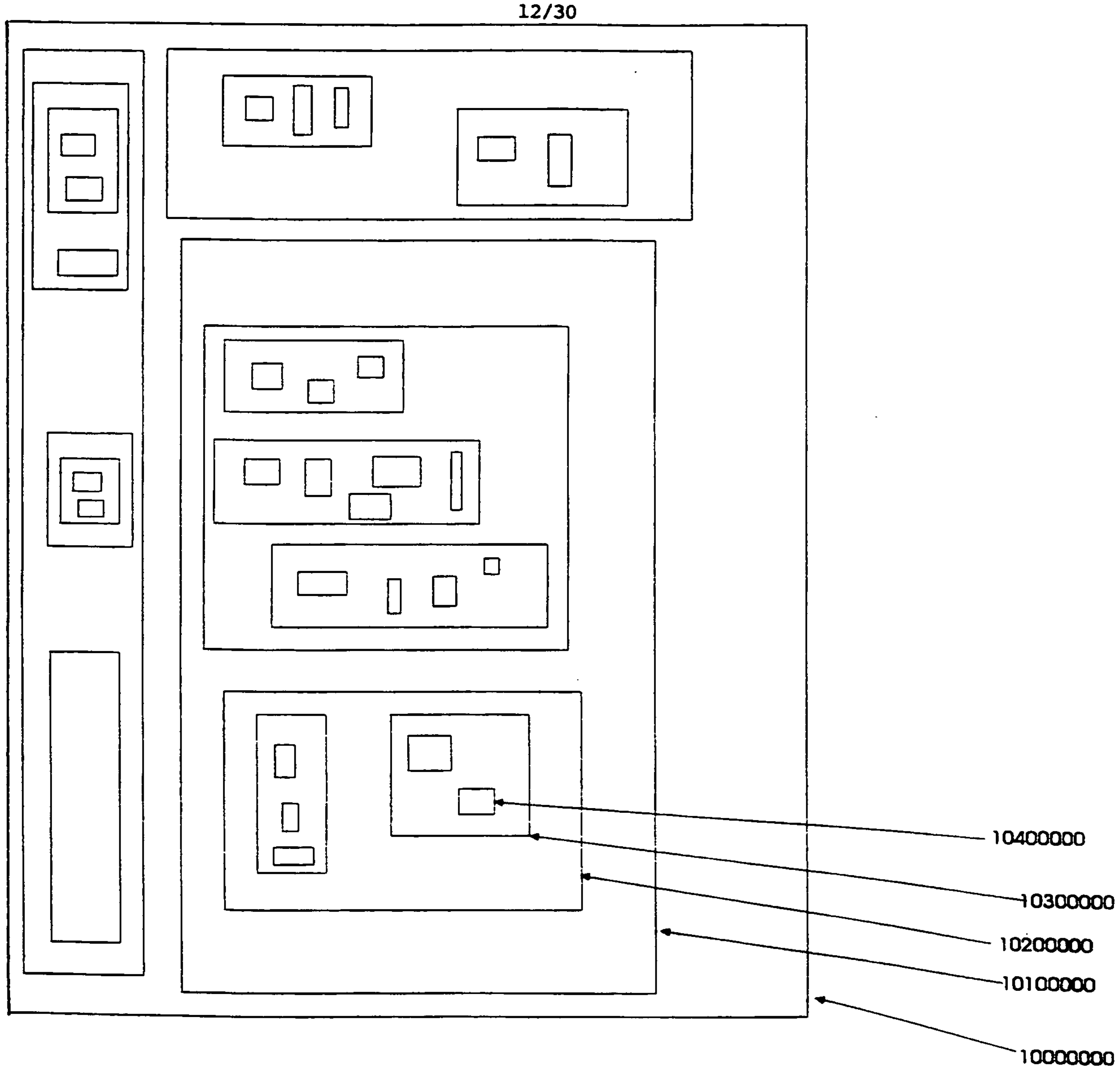


FIG. 3 B

13/30

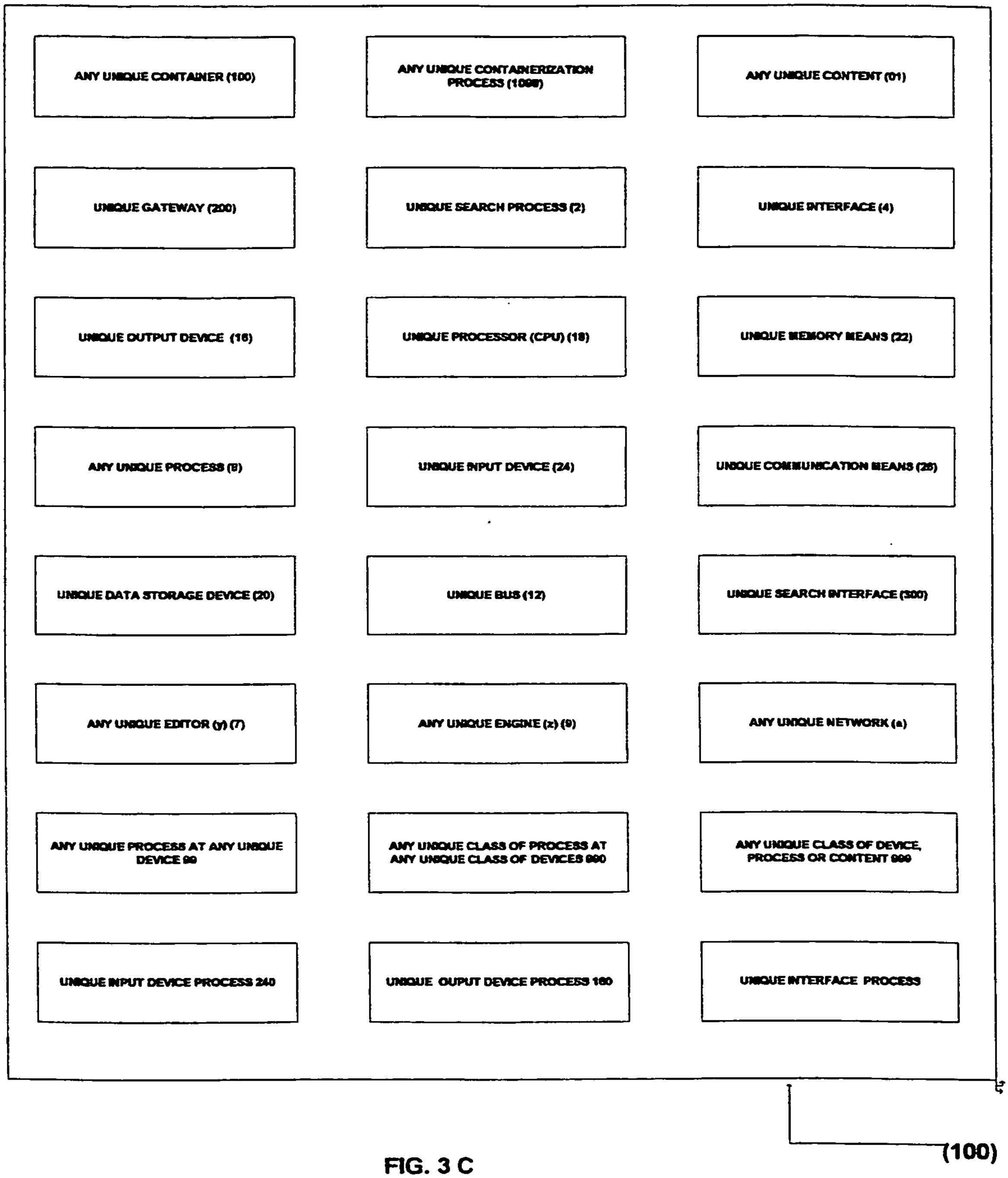

ANY UNIQUE CONTAINER (100)
ANY UNIQUE CONTAINERIZATION PROCESS (1000)
ANY UNIQUE CONTENT (01)
UNIQUE GATEWAY (200)
UNIQUE SEARCH PROCESS (2)
UNIQUE INTERFACE (4)
UNIQUE OUTPUT DEVICE (16)
UNIQUE PROCESSOR (CPU) (18)
UNIQUE MEMORY MEANS (22)
ANY UNIQUE PROCESS (8)
UNIQUE INPUT DEVICE (24)
UNIQUE COMMUNICATION MEANS (26)
UNIQUE DATA STORAGE DEVICE (20)
UNIQUE BUS (12)
UNIQUE SEARCH INTERFACE (300)
ANY UNIQUE EDITOR (y) (7)
ANY UNIQUE ENGINE (z) (9)
ANY UNIQUE NETWORK (a)
ANY UNIQUE PROCESS AT ANY UNIQUE DEVICE 99
ANY UNIQUE CLASS OF PROCESS AT ANY UNIQUE CLASS OF DEVICES 990
ANY UNIQUE CLASS OF DEVICE, PROCESS OR CONTENT 999
UNIQUE INPUT DEVICE PROCESS 240
UNIQUE OUPUT DEVICE PROCESS 160
UNIQUE INTERFACE PROCESS
(100)

FIG. 3 C

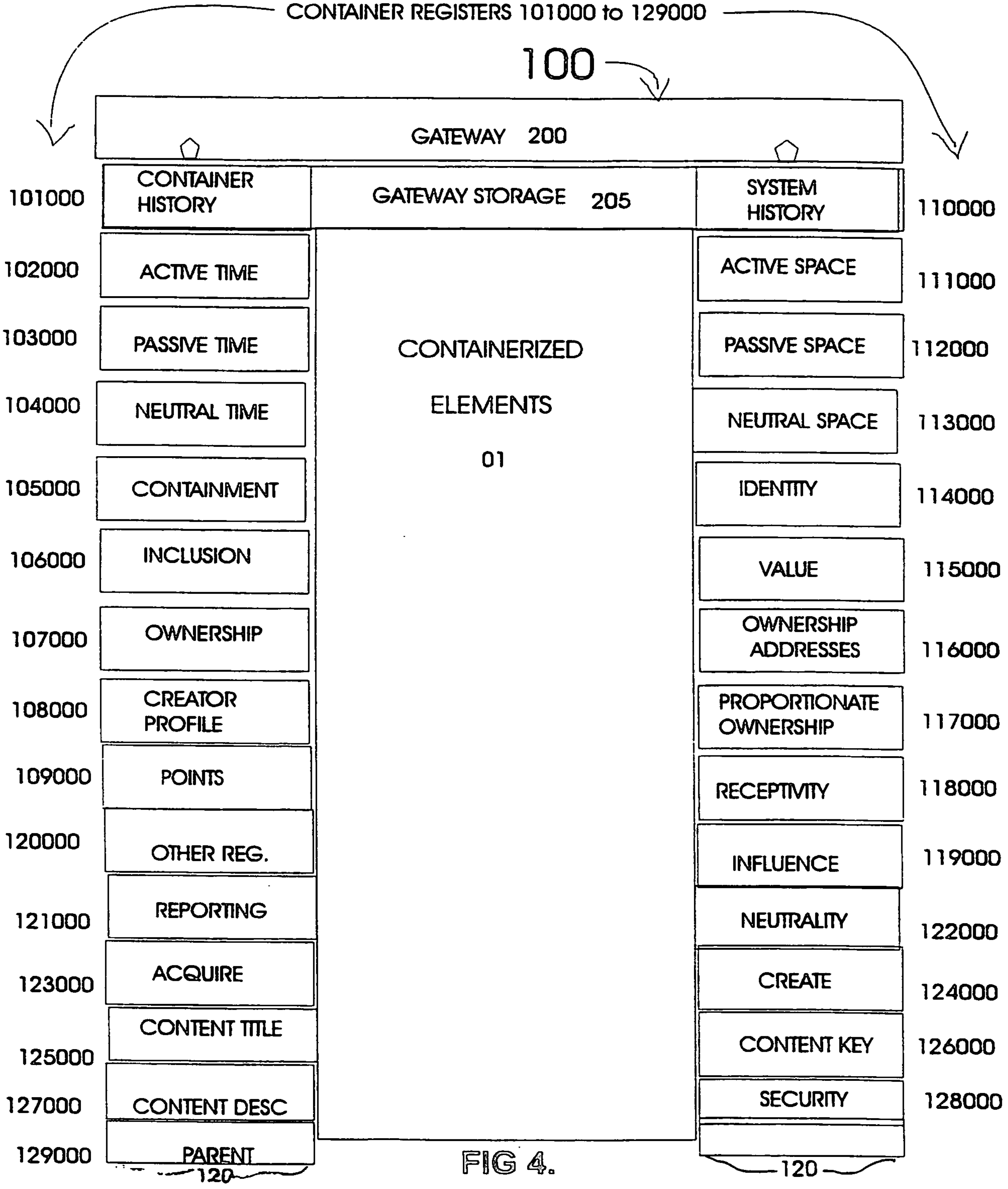

CONTAINER REGISTERS 101000 to 129000
100
GATEWAY 200
101000
CONTAINER HISTORY
GATEWAY STORAGE 205
SYSTEM HISTORY
110000
102000
ACTIVE TIME
ACTIVE SPACE
111000
103000
PASSIVE TIME
CONTAINERIZED ELEMENTS 01
PASSIVE SPACE
112000
104000
NEUTRAL TIME
NEUTRAL SPACE
113000
105000
CONTAINMENT
IDENTITY
114000
106000
INCLUSION
VALUE
115000
107000
OWNERSHIP
OWNERSHIP ADDRESSES
116000
108000
CREATOR PROFILE
PROPORTIONATE OWNERSHIP
117000
109000
POINTS
RECEPTIVITY
118000
120000
OTHER REG.
INFLUENCE
119000
121000
REPORTING
NEUTRALITY
122000
123000
ACQUIRE
CREATE
124000
125000
CONTENT TITLE
CONTENT KEY
126000
127000
CONTENT DESC
SECURITY
128000
129000
PARENT
120
120

FIG 4.

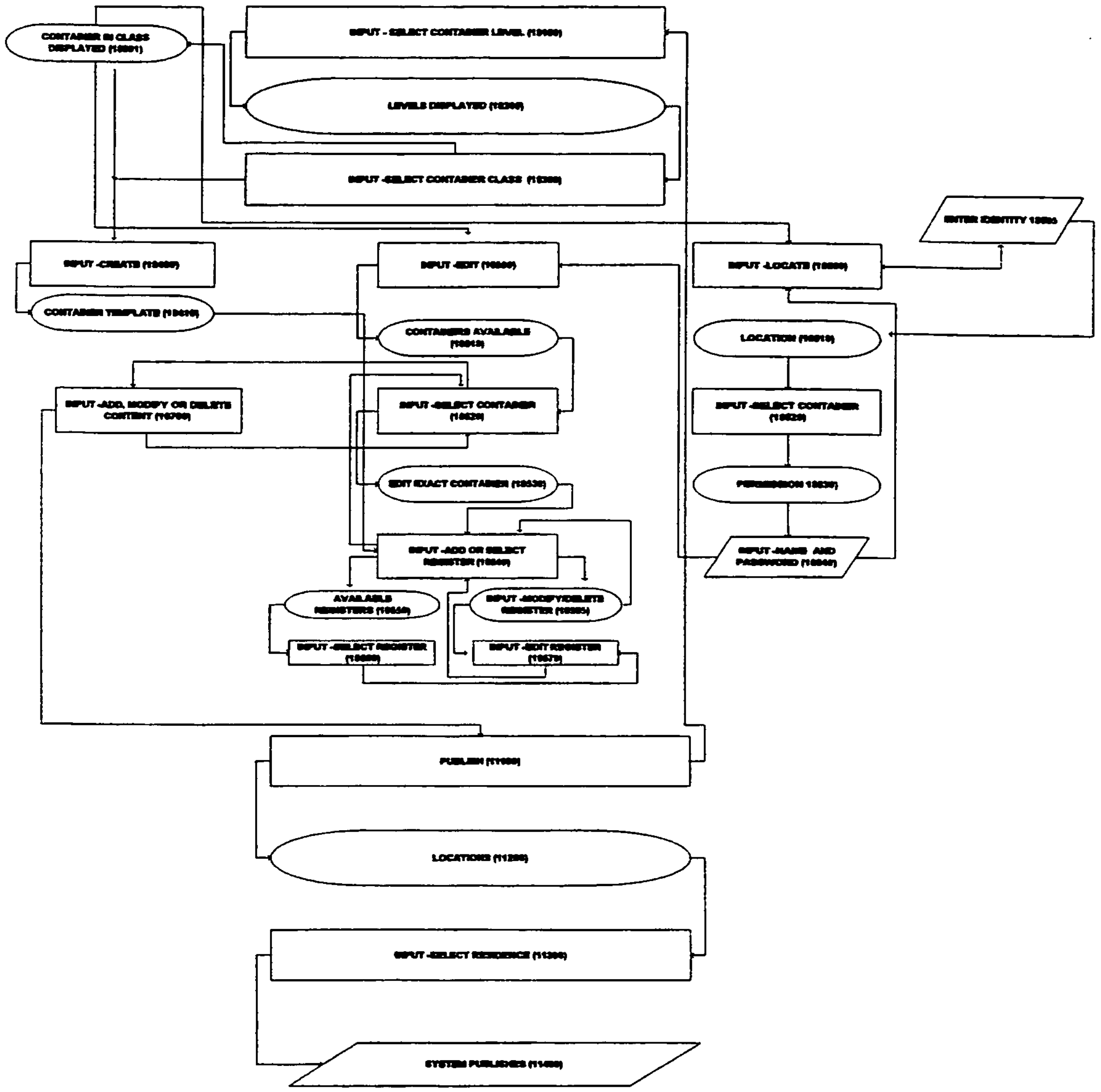

CONTAINER IN CLASS DISPLATED (10001)
INPUT - SELECT CONTAINER LEVEL (10100)
LEVELS DISPLAYED (10200)
INPUT -SELECT CONTAINER CLASS (10300)
ENTER IDENTITY 10805
INPUT -CREATE (10400)
INPUT -EDIT (10500)
INPUT -LOCATE (10800)
CONTAINER TEMPLATE (10410)
CONTAINERS AVAILABLE (10510)
LOCATION (10810)
INPUT -ADD, MODIFY OR DELETE CONTENT (10700)
INPUT -SELECT CONTAINER (10520)
INPUT -SELECT CONTAINER (10820)
EDIT EXACT CONTAINER (10530)
PERMISSION 10830
INPUT -ADD OR SELECT REGISTER (10540)
INPUT -NAME AND PASSWORD (10840)
AVAILABLE REGISTERS (10550)
INPUT -MODIFY/DELETE REGISTER (10565)
INPUT -SELECT REGISTER (10560)
INPUT -EDIT REGISTER (10570)
PUBLISH (11000)
LOCATIONS (11200)
INPUT -SELECT RESIDENCE (11300)
SYSTEM PUBLISHES (11400)

FIG. 5

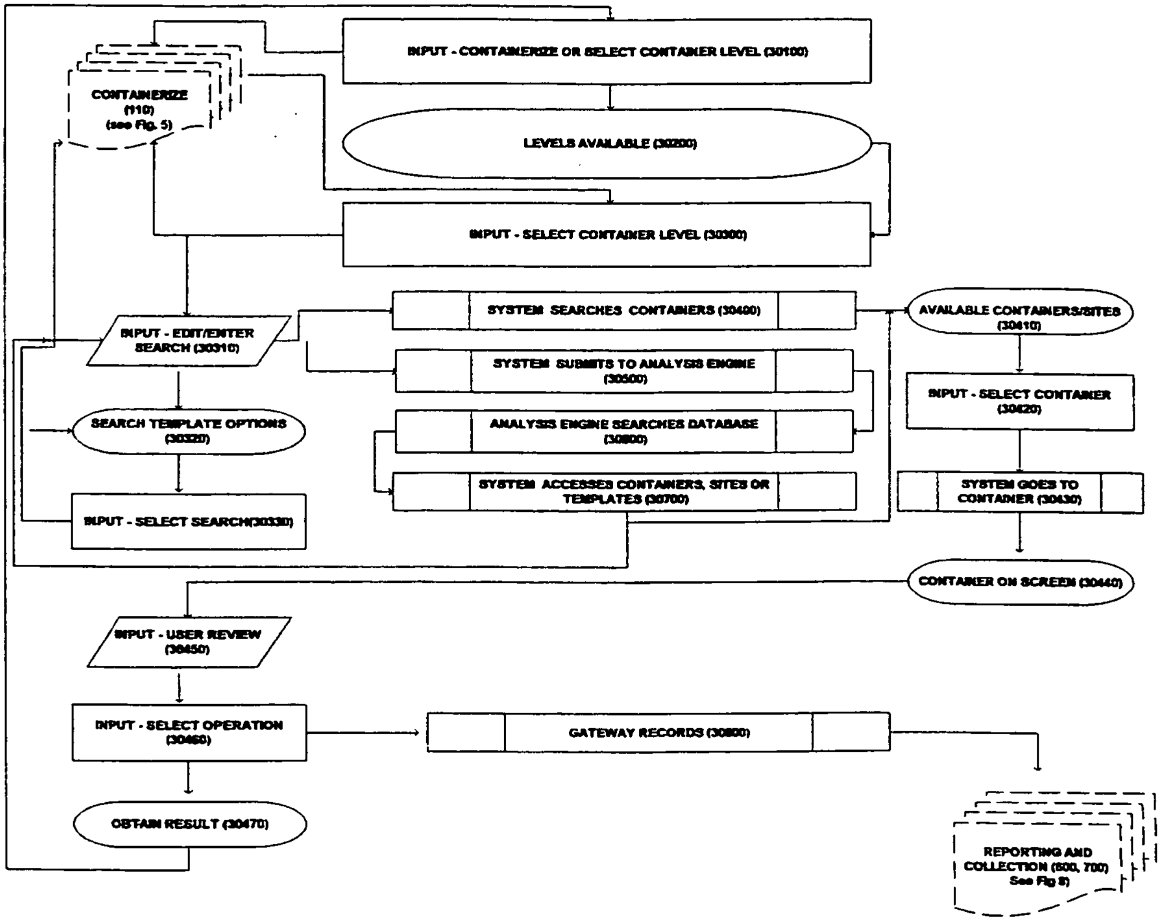

INPUT - CONTAINERIZE OR SELECT CONTAINER LEVEL (30100)
CONTAINERIZE (110) (see Fig. 5)
LEVELS AVAILABLE (30200)
INPUT - SELECT CONTAINER LEVEL (30300)
INPUT - EDIT/ENTER SEARCH (30310)
SYSTEM SEARCHES CONTAINERS (30400)
AVAILABLE CONTAINERS/SITES (30410)
SYSTEM SUBMITS TO ANALYSIS ENGINE (30500)
INPUT - SELECT CONTAINER (30420)
SEARCH TEMPLATE OPTIONS (30320)
ANALYSIS ENGINE SEARCHES DATABASE (30600)
SYSTEM ACCESSES CONTAINERS, SITES OR TEMPLATES (30700)
SYSTEM GOES TO CONTAINER (30430)
INPUT - SELECT SEARCH(30330)
CONTAINER ON SCREEN (30440)
INPUT - USER REVIEW (30450)
INPUT - SELECT OPERATION (30460)
GATEWAY RECORDS (30800)
OBTAIN RESULT (30470)
REPORTING AND COLLECTION (600, 700) See Fig 8)

FIG. 6

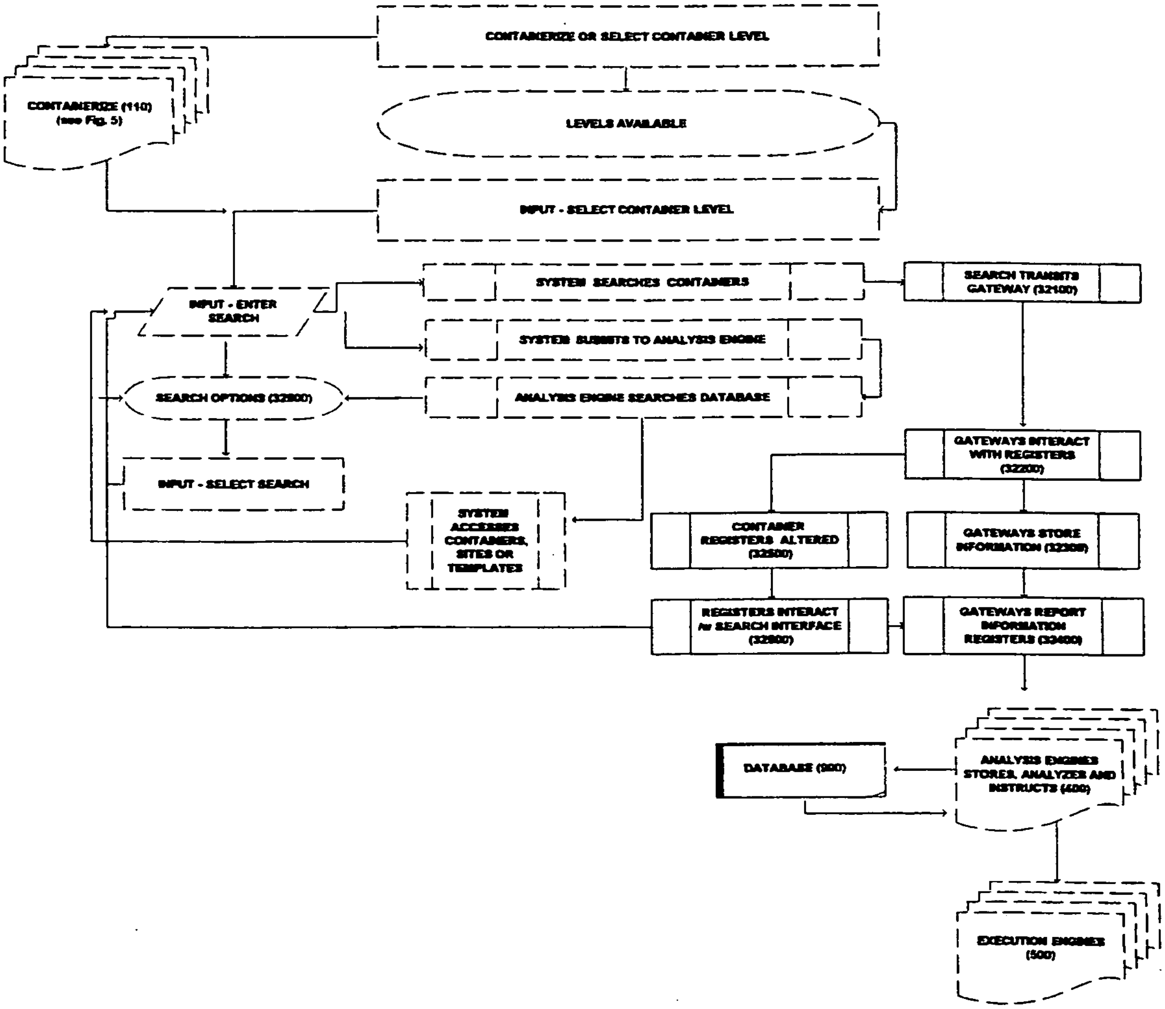

CONTAINERIZE OR SELECT CONTAINER LEVEL
CONTAINERIZE (110) (see Fig. 5)
LEVELS AVAILABLE
INPUT - SELECT CONTAINER LEVEL
INPUT - ENTER SEARCH
SYSTEM SEARCHES CONTAINERS
SEARCH TRANSITS GATEWAY (32100)
SYSTEM SUBMITS TO ANALYSIS ENGINE
SEARCH OPTIONS (32800)
ANALYSIS ENGINE SEARCHES DATABASE
INPUT - SELECT SEARCH
GATEWAYS INTERACT WITH REGISTERS (32200)
SYSTEM ACCESSES CONTAINERS, SITES OR TEMPLATES
CONTAINER REGISTERS ALTERED (32500)
GATEWAYS STORE INFORMATION (32300)
REGISTERS INTERACT /w SEARCH INTERFACE (32600)
GATEWAYS REPORT INFORMATION REGISTERS (32400)
DATABASE (900)
ANALYSIS ENGINES STORES, ANALYZES AND INSTRUCTS (400)
EXECUTION ENGINES (500)

FIG. 7

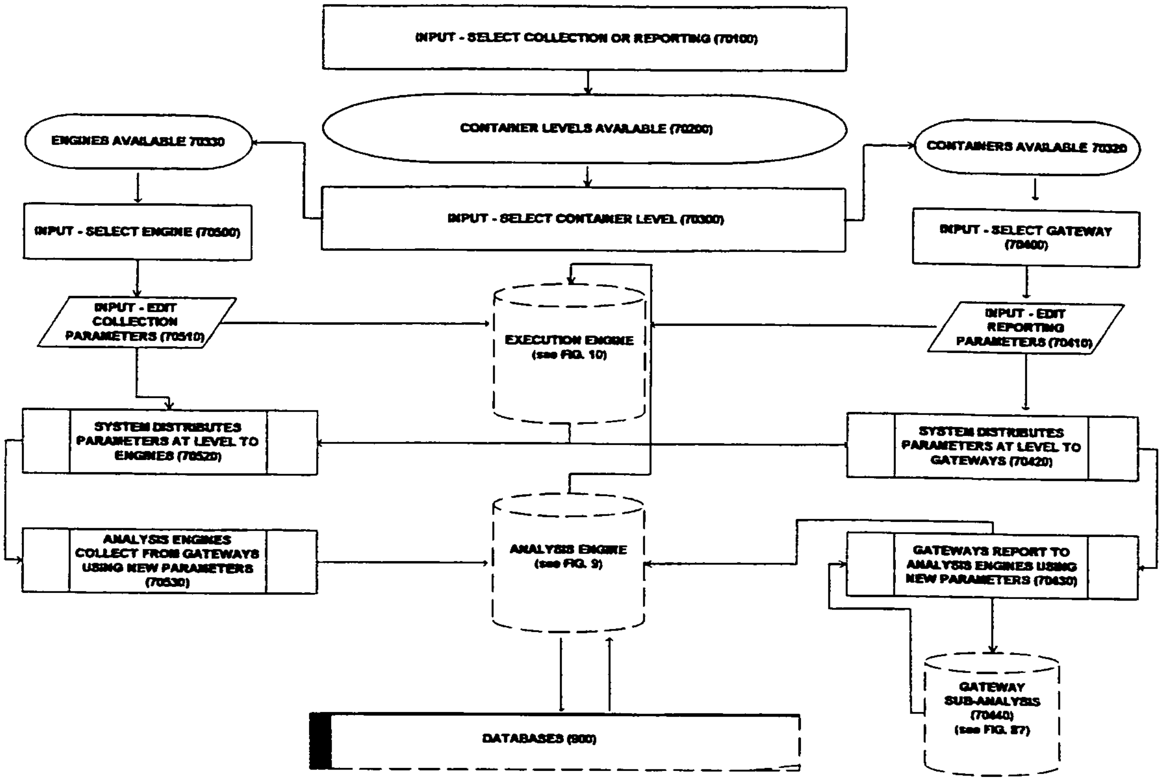

INPUT - SELECT COLLECTION OR REPORTING (70100)
CONTAINER LEVELS AVAILABLE (70200)
ENGINES AVAILABLE 70330
CONTAINERS AVAILABLE 70320
INPUT - SELECT CONTAINER LEVEL (70300)
INPUT - SELECT ENGINE (70500)
INPUT - SELECT GATEWAY (70400)
INPUT - EDIT COLLECTION PARAMETERS (70510)
EXECUTION ENGINE (see FIG. 10)
INPUT - EDIT REPORTING PARAMETERS (70410)
SYSTEM DISTRIBUTES PARAMETERS AT LEVEL TO ENGINES (70520)
SYSTEM DISTRIBUTES PARAMETERS AT LEVEL TO GATEWAYS (70420)
ANALYSIS ENGINES COLLECT FROM GATEWAYS USING NEW PARAMETERS (70530)
ANALYSIS ENGINE (see FIG. 9)
GATEWAYS REPORT TO ANALYSIS ENGINES USING NEW PARAMETERS (70430)
GATEWAY SUB-ANALYSIS (70440) (see FIG. 87)
DATABASES (900)

FIG. 8

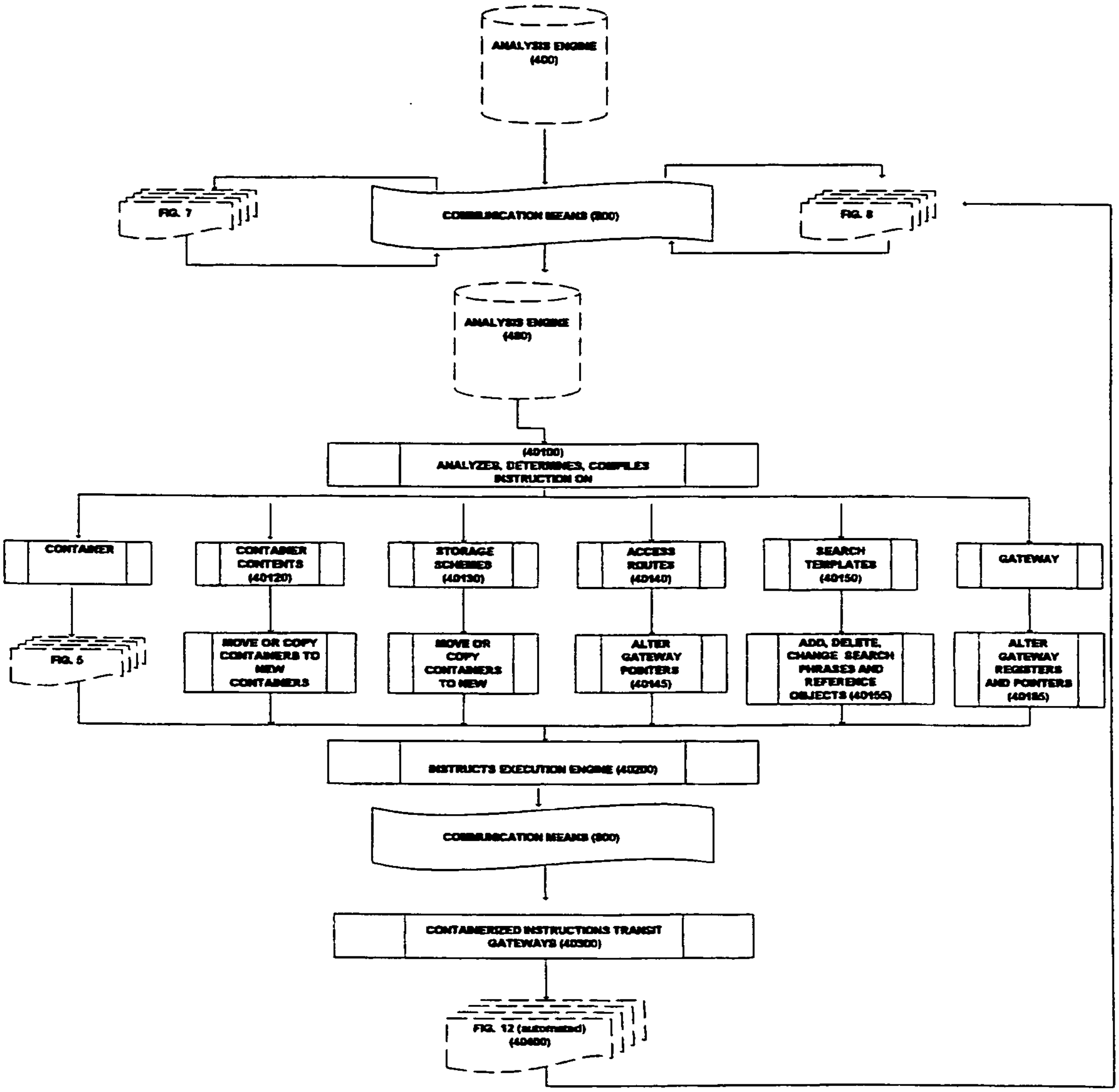

ANALYSIS ENGINE (400)
FIG. 7
COMMUNICATION MEANS (300)
FIG. 8
ANALYSIS ENGINE (480)
(40100) ANALYZES, DETERMINES, COMPILES INSTRUCTION ON
CONTAINER
CONTAINER CONTENTS (40120)
STORAGE SCHEMES (40130)
ACCESS ROUTES (40140)
SEARCH TEMPLATES (40150)
GATEWAY
FIG. 5
MOVE OR COPY CONTAINERS TO NEW CONTAINERS
MOVE OR COPY CONTAINERS TO NEW
ALTER GATEWAY POINTERS (40145)
ADD, DELETE, CHANGE SEARCH PHRASES AND REFERENCE OBJECTS (40155)
ALTER GATEWAY REGISTERS AND POINTERS (40185)
INSTRUCTS EXECUTION ENGINE (40200)
COMMUNICATION MEANS (300)
CONTAINERIZED INSTRUCTIONS TRANSIT GATEWAYS (40300)
FIG. 12 (automated) (40400)

FIG. 9

EXECUTION ENGINE

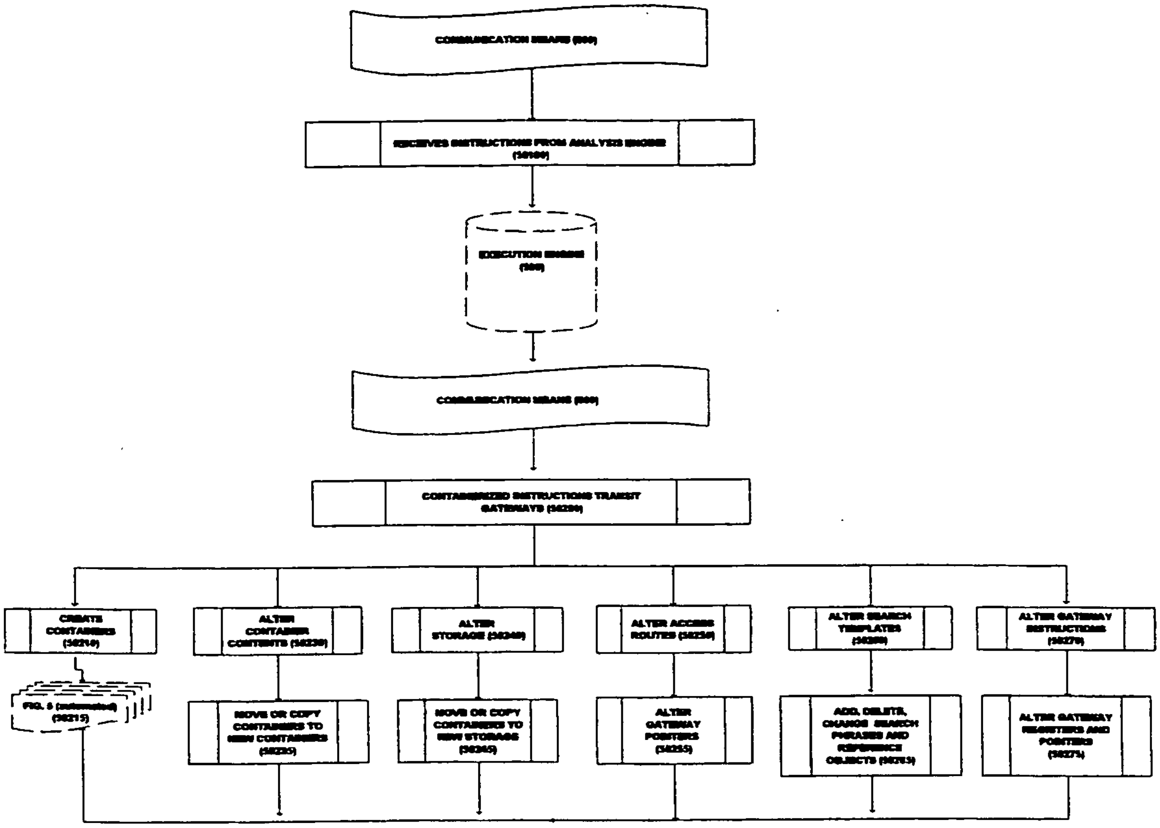

COMMUNICATION MEANS (800)
RECEIVES INSTRUCTIONS FROM ANALYSIS ENGINE (34100)
EXECUTION ENGINE (300)
COMMUNICATION MEANS (800)
CONTAINERIZED INSTRUCTIONS TRANSIT GATEWAYS (34290)
CREATE CONTAINERS (34210)
FIG. 5 (automated) (34215)
ALTER CONTAINER CONTENTS (34220)
MOVE OR COPY CONTAINERS TO NEW CONTAINERS (34225)
ALTER STORAGE (34240)
MOVE OR COPY CONTAINERS TO NEW STORAGE (34245)
ALTER ACCESS ROUTES (34250)
ALTER GATEWAY POINTERS (34255)
ALTER SEARCH TEMPLATES (34260)
ADD, DELETE, CHANGE SEARCH PHRASES AND REFERENCE OBJECTS (34265)
ALTER GATEWAY INSTRUCTIONS (34270)
ALTER GATEWAY REGISTERS AND POINTERS (34275)

FIG.10

GATEWAY EDITOR

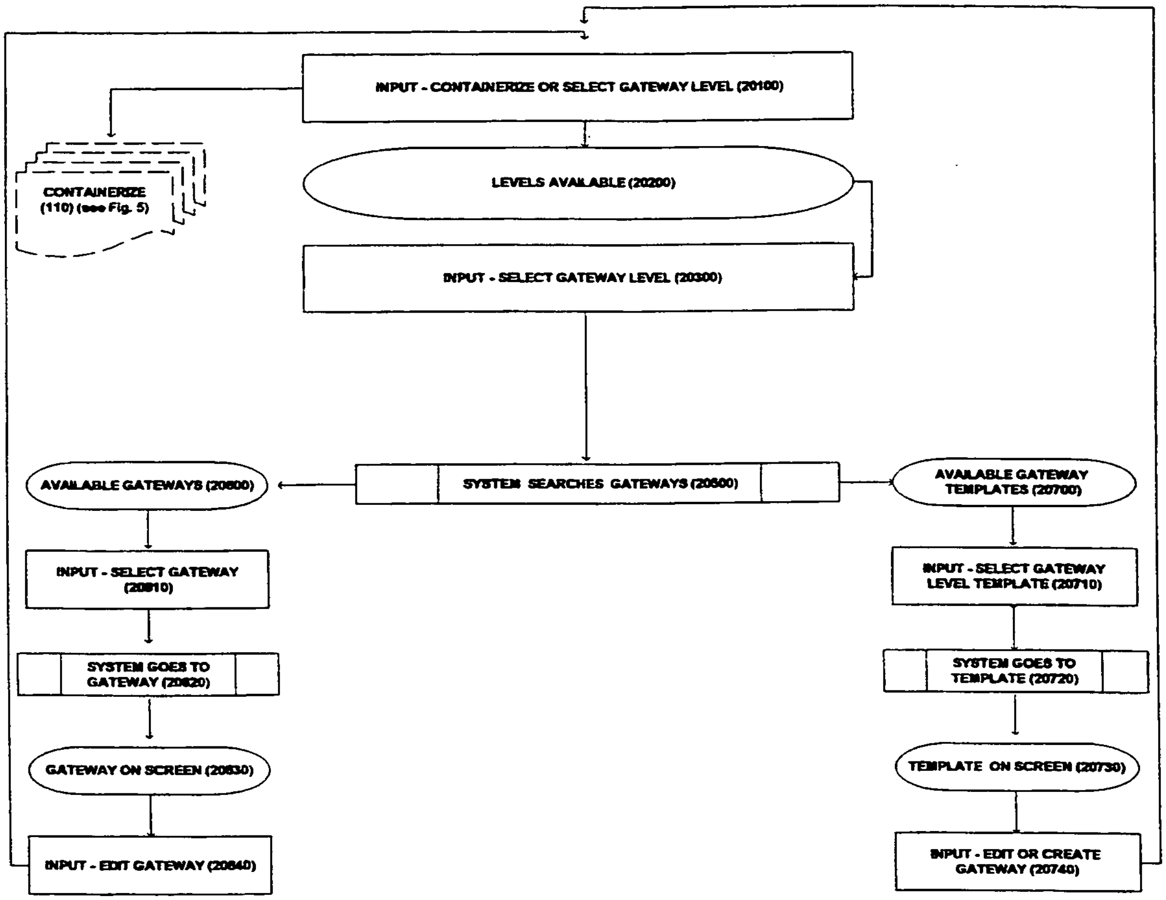

INPUT - CONTAINERIZE OR SELECT GATEWAY LEVEL (20100)
CONTAINERIZE (110) (see Fig. 5)
LEVELS AVAILABLE (20200)
INPUT - SELECT GATEWAY LEVEL (20300)
SYSTEM SEARCHES GATEWAYS (20600)
AVAILABLE GATEWAYS (20800)
INPUT - SELECT GATEWAY (20810)
SYSTEM GOES TO GATEWAY (20820)
GATEWAY ON SCREEN (20830)
INPUT - EDIT GATEWAY (20840)
AVAILABLE GATEWAY TEMPLATES (20700)
INPUT - SELECT GATEWAY LEVEL TEMPLATE (20710)
SYSTEM GOES TO TEMPLATE (20720)
TEMPLATE ON SCREEN (20730)
INPUT - EDIT OR CREATE GATEWAY (20740)

FIG. 11

GATEWAY PROCESS

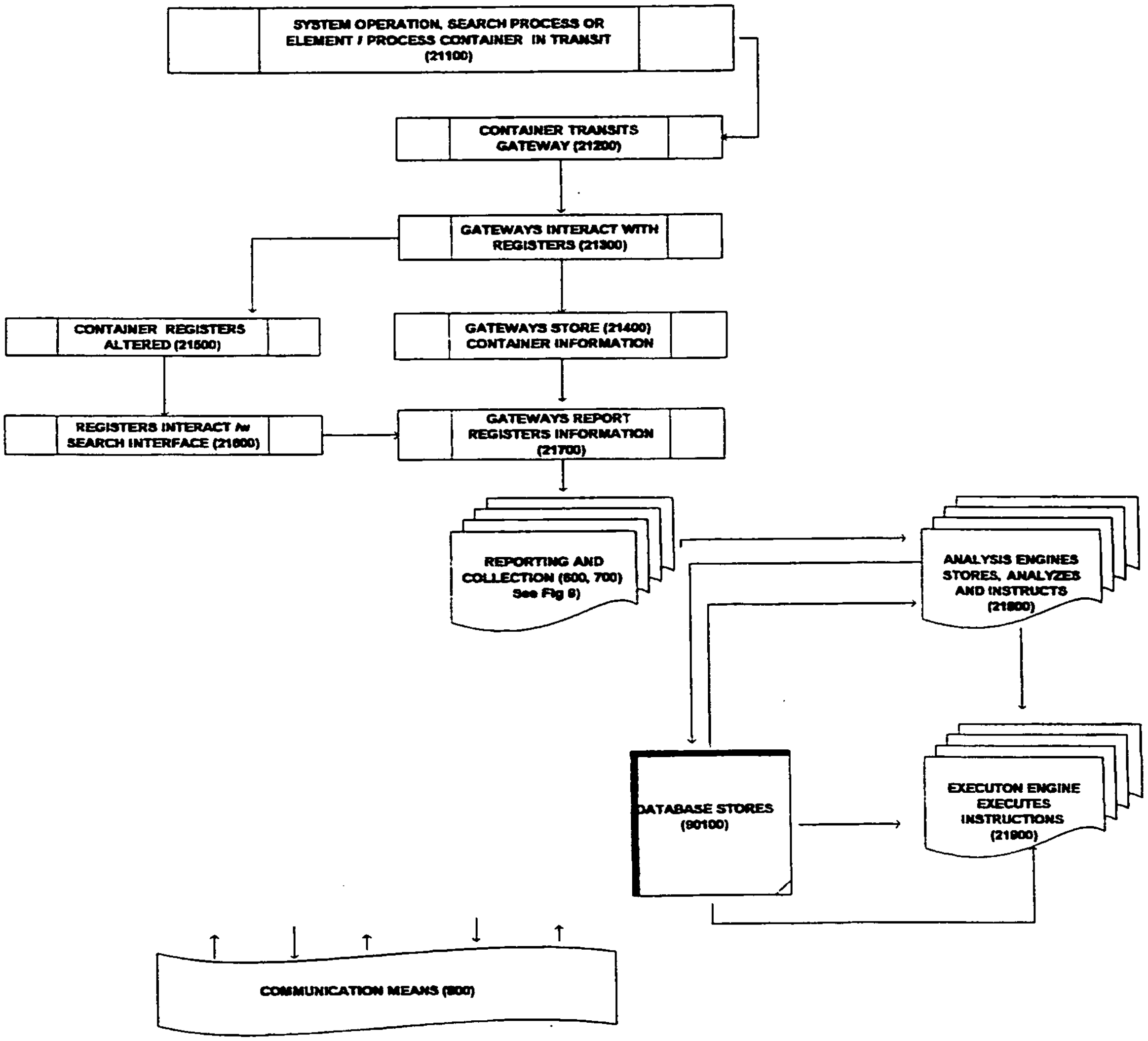

SYSTEM OPERATION, SEARCH PROCESS OR ELEMENT / PROCESS CONTAINER IN TRANSIT (21100)
CONTAINER TRANSITS GATEWAY (21200)
GATEWAYS INTERACT WITH REGISTERS (21300)
CONTAINER REGISTERS ALTERED (21500)
GATEWAYS STORE (21400) CONTAINER INFORMATION
REGISTERS INTERACT /w SEARCH INTERFACE (21600)
GATEWAYS REPORT REGISTERS INFORMATION (21700)
REPORTING AND COLLECTION (600, 700) See Fig 9)
ANALYSIS ENGINES STORES, ANALYZES AND INSTRUCTS (21800)
DATABASE STORES (90100)
EXECUTON ENGINE EXECUTES INSTRUCTIONS (21900)
COMMUNICATION MEANS (800)

FIG. 12

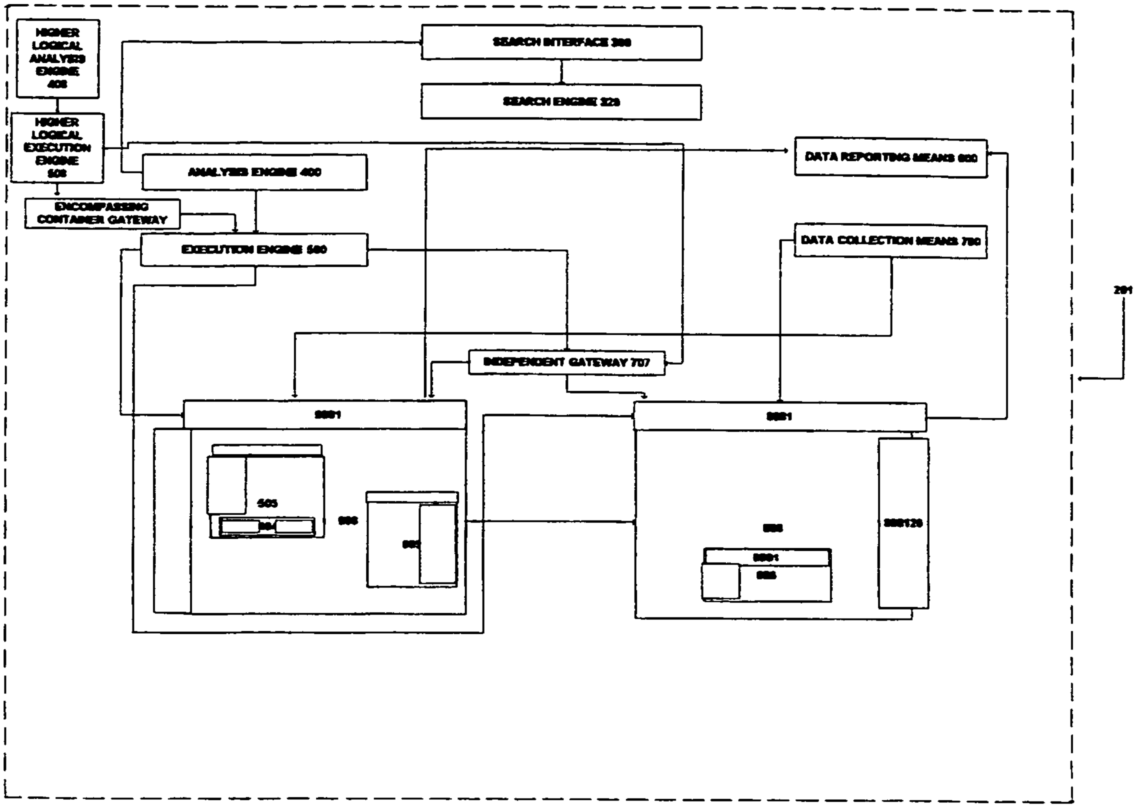

HIGHER LOGICAL ANALYZER ENGINE 408
SEARCH INTERFACE 300
SEARCH ENGINE 320
HIGHER LOGICAL EXECUTION ENGINE 508
ANALYZER ENGINE 400
DATA REPORTING MEANS 800
ENCOMPASSING CONTAINER GATEWAY
EXECUTION ENGINE 500
DATA COLLECTION MEANS 700
201
INDEPENDENT GATEWAY 707
5001
5001
505
504
506
506
506
5001
506
500120

FIG. 13 A

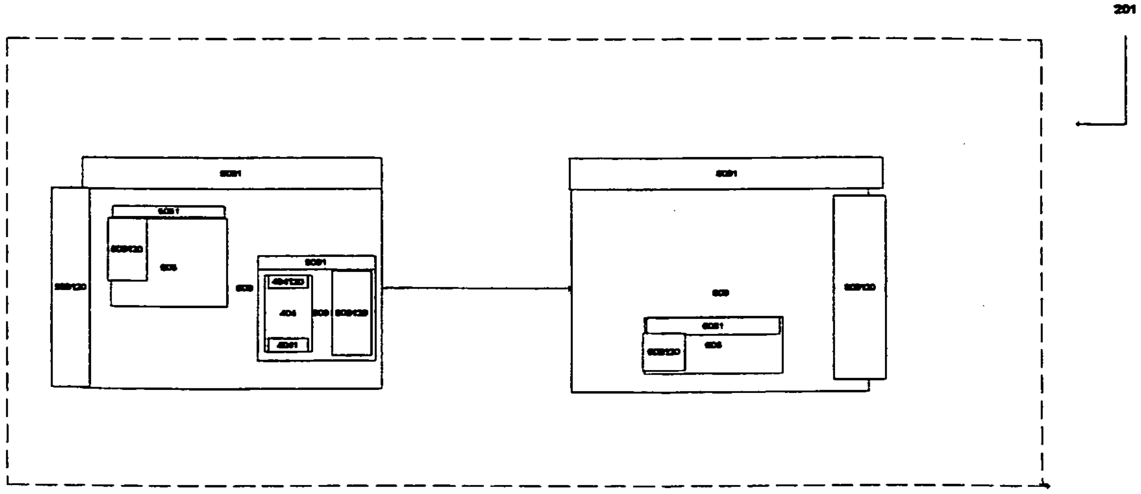


FIG. 13 B

09/284113

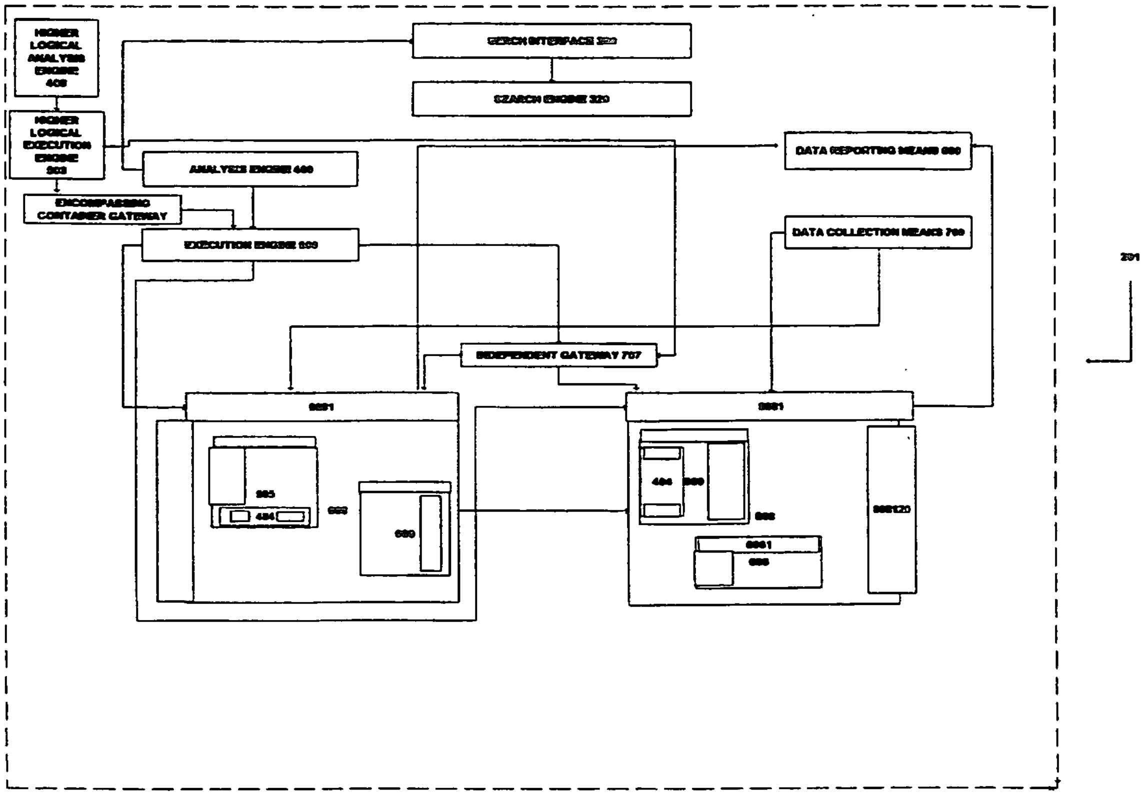


FIG. 13 C

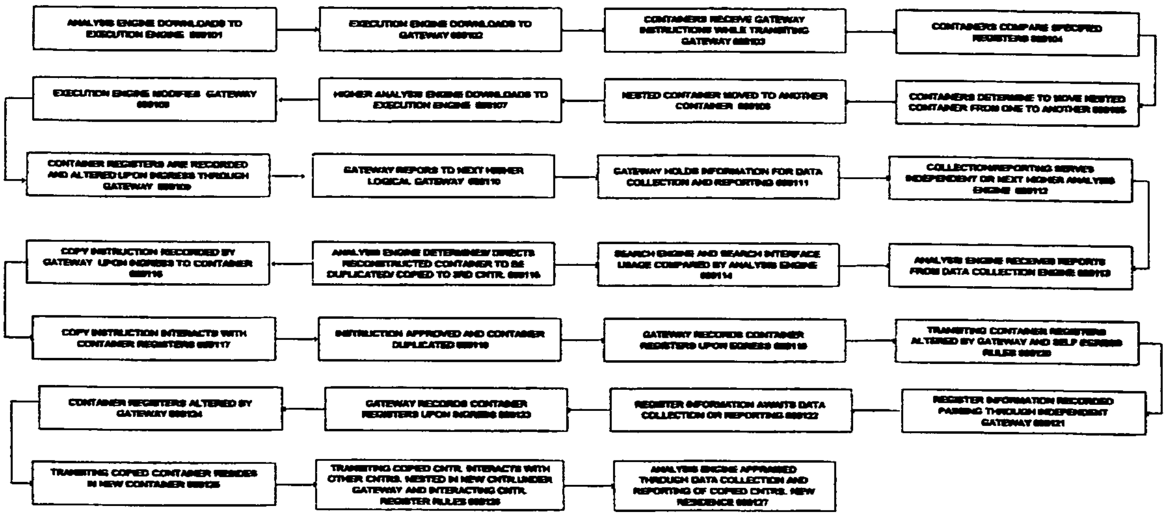

ANALYSIS ENGINE DOWNLOADS TO EXECUTION ENGINE 888101
EXECUTION ENGINE DOWNLOADS TO GATEWAY 888102
CONTAINERS RECEIVE GATEWAY INSTRUCTIONS WHILE TRANSITING GATEWAY 888103
CONTAINERS COMPARE SPECIFIED REGISTERS 888104
CONTAINERS DETERMINE TO MOVE NESTED CONTAINER FROM ONE TO ANOTHER 888105
NESTED CONTAINER MOVED TO ANOTHER CONTAINER 888106
HIGHER ANALYSIS ENGINE DOWNLOADS TO EXECUTION ENGINE 888107
EXECUTION ENGINE MODIFIES GATEWAY 888108
CONTAINER REGISTERS ARE RECORDED AND ALTERED UPON INGRESS THROUGH GATEWAY 888109
GATEWAY REPORS TO NEXT HIGHER LOGICAL GATEWAY 888110
GATEWAY HOLDS INFORMATION FOR DATA COLLECTION AND REPORTING 888111
COLLECTION/REPORTING SERVES INDEPENDENT OR NEXT HIGHER ANALYSIS ENGINE 888112
ANALYSIS ENGINE RECEIVES REPORTS FROM DATA COLLECTION ENGINE 888113
SEARCH ENGINE AND SEARCH INTERFACE USAGE COMPARED BY ANALYSIS ENGINE 888114
ANALYSIS ENGINE DETERMINES/ DIRECTS RECONSTRUCTED CONTAINER TO BE DUPLICATED/ COPIED TO 3RD CNTR. 888115
COPY INSTRUCTION RECORDED BY GATEWAY UPON INGRESS TO CONTAINER 888116
COPY INSTRUCTION INTERACTS WITH CONTAINER REGISTERS 888117
INSTRUCTION APPROVED AND CONTAINER DUPLICATED 888118
GATEWAY RECORDS CONTAINER REGISTERS UPON EGRESS 888119
TRANSITING CONTAINER REGISTERS ALTERED BY GATEWAY AND SELF ESCROW RULES 888120
REGISTER INFORMATION RECORDED PASSING THROUGH INDEPENDENT GATEWAY 888121
REGISTER INFORMATION AWAITS DATA COLLECTION OR REPORTING 888122
GATEWAY RECORDS CONTAINER REGISTERS UPON INGRESS 888123
CONTAINER REGISTERS ALTERED BY GATEWAY 888124
TRANSITING COPIED CONTAINER RESIDES IN NEW CONTAINER 888125
TRANSITING COPIED CNTR. INTERACTS WITH OTHER CNTRS. NESTED IN NEW CNTR.UNDER GATEWAY AND INTERACTING CNTR. REGISTER RULES 888126
ANALYSIS ENGINE APPRAISED THROUGH DATA COLLECTION AND REPORTING OF COPIED CNTRS. NEW RESIDENCE 888127

FIG. 13 D

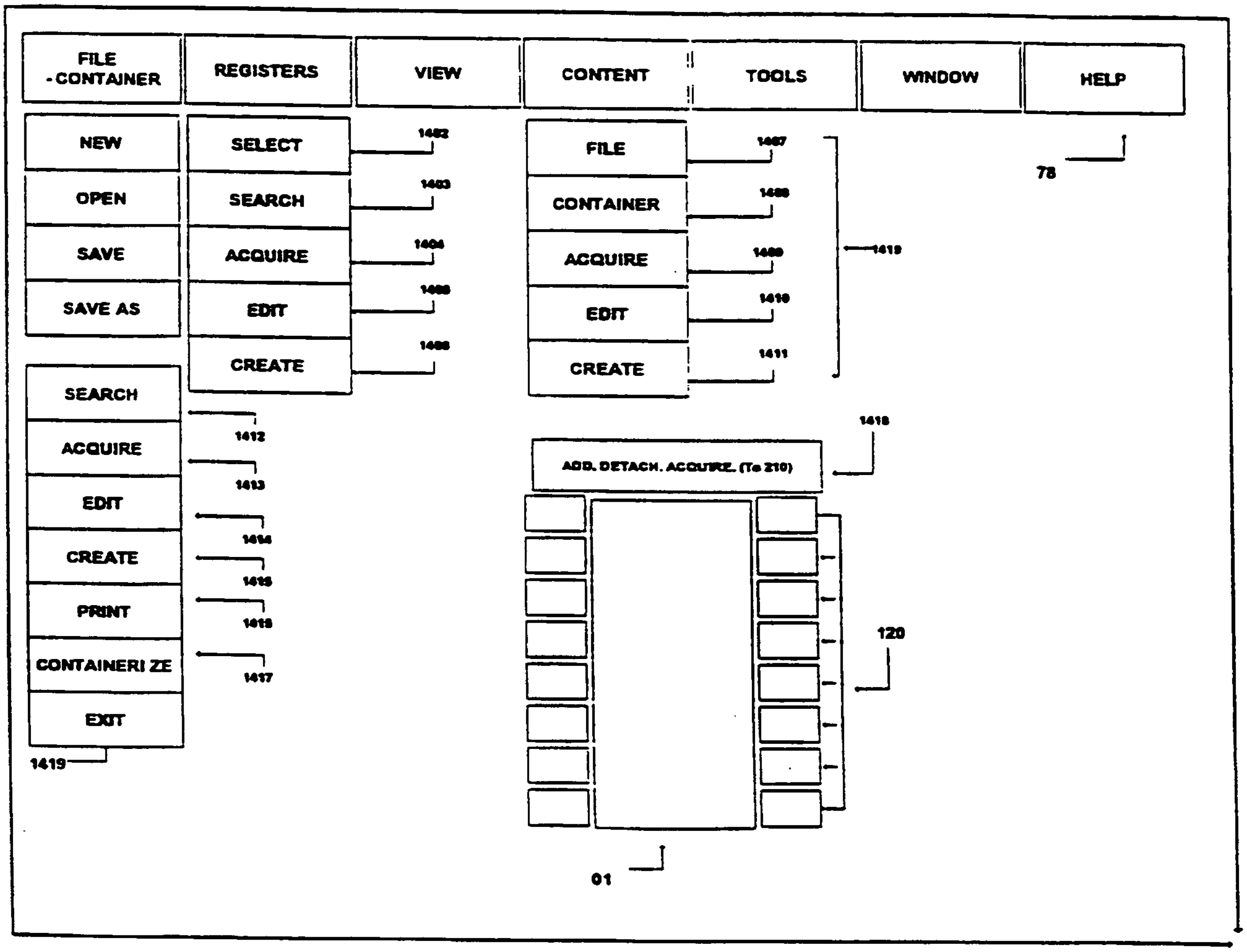

FILE - CONTAINER
REGISTERS
VIEW
CONTENT
TOOLS
WINDOW
HELP
NEW
OPEN
SAVE
SAVE AS
SELECT
SEARCH
ACQUIRE
EDIT
CREATE
1402
1403
1404
1405
1406
FILE
CONTAINER
ACQUIRE
EDIT
CREATE
1407
1408
1409
1410
1411
1419
78
SEARCH
ACQUIRE
EDIT
CREATE
PRINT
CONTAINERI ZE
EXIT
1412
1413
1414
1415
1416
1417
1419
1418
ADD. DETACH. ACQUIRE. (To 210)
120
01

FIG. 14

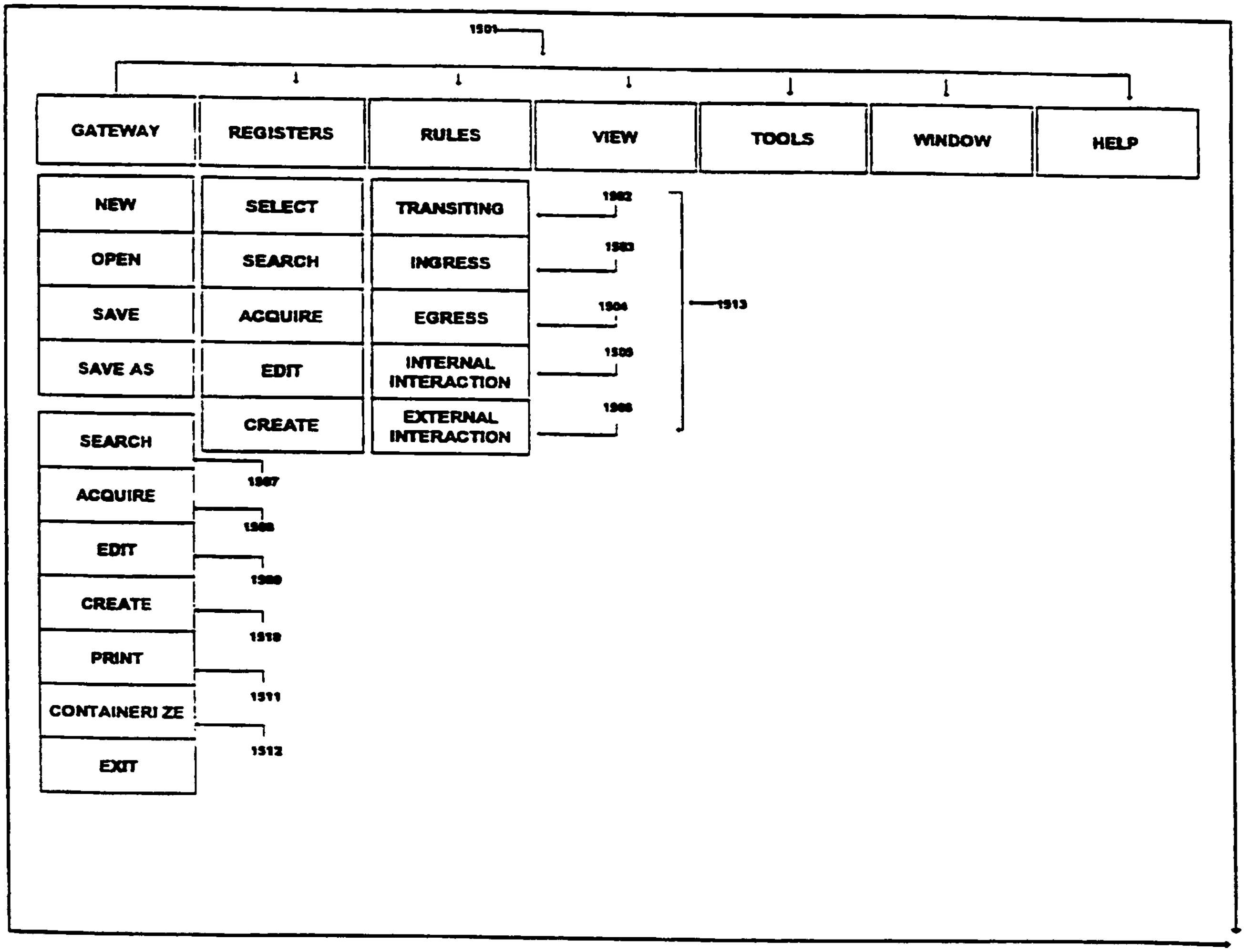

1501
GATEWAY
REGISTERS
RULES
VIEW
TOOLS
WINDOW
HELP
NEW
OPEN
SAVE
SAVE AS
SEARCH
ACQUIRE
EDIT
CREATE
PRINT
CONTAINERI ZE
EXIT
SELECT
SEARCH
ACQUIRE
EDIT
CREATE
TRANSITING
INGRESS
EGRESS
INTERNAL INTERACTION
EXTERNAL INTERACTION
1502
1503
1504
1505
1506
1513
1507
1508
1509
1510
1511
1512

FIG. 15

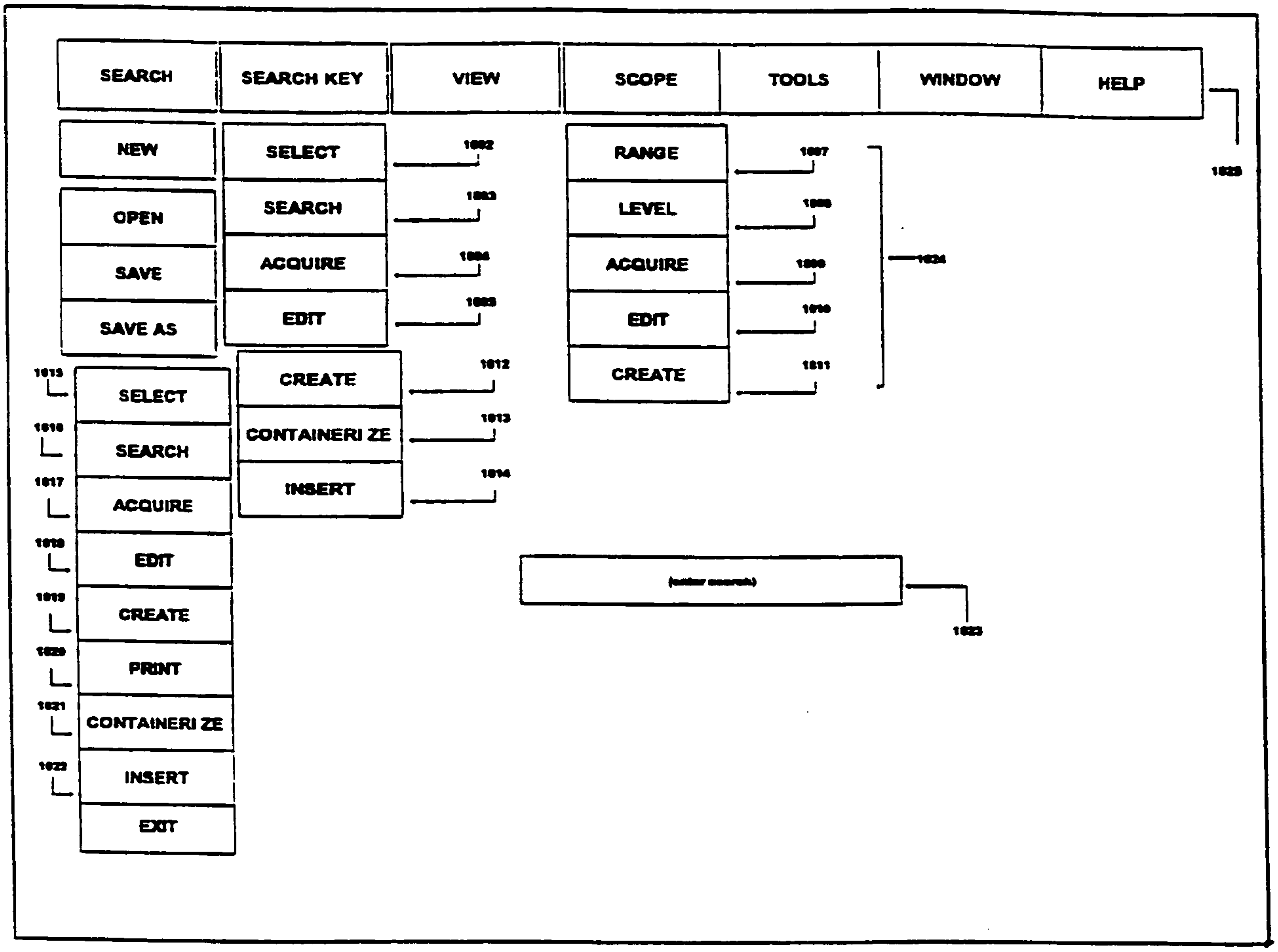

SEARCH
SEARCH KEY
VIEW
SCOPE
TOOLS
WINDOW
HELP
1625
NEW
OPEN
SAVE
SAVE AS
SELECT
SEARCH
ACQUIRE
EDIT
1602
1603
1604
1605
CREATE
CONTAINERI ZE
INSERT
1612
1613
1614
RANGE
LEVEL
ACQUIRE
EDIT
CREATE
1607
1608
1609
1610
1611
1624
1615
SELECT
1616
SEARCH
1617
ACQUIRE
1618
EDIT
1619
CREATE
1620
PRINT
1621
CONTAINERI ZE
1622
INSERT
EXIT
(enter search)
1623

FIG. 16

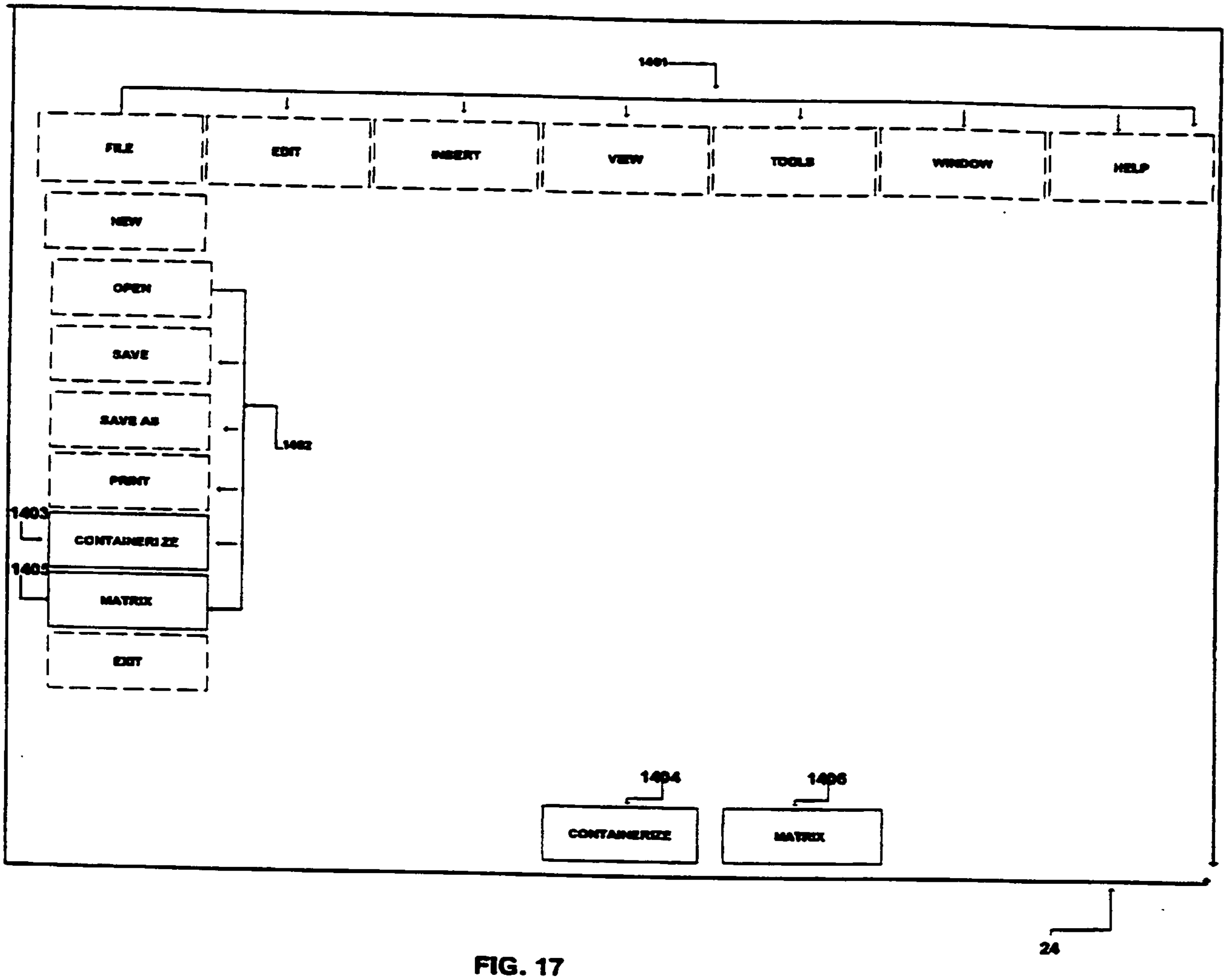

1401
FILE
EDIT
INSERT
VIEW
TOOLS
WINDOW
HELP
NEW
OPEN
SAVE
SAVE AS
PRINT
1403
CONTAINERI ZE
1405
MATRIX
EXIT
1402
1404
CONTAINERIZE
1406
MATRIX
24

FIG. 17

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to computer systems in a multi-user mainframe or mini computer system, a client server network, or in local, wide area or public networks, and in particular, to computer networks for creating and manipulating information containers with dynamic interactive registers in a computer, media or publishing network, in order to manufacture information on, upgrade the utility of, and develop intelligence in, a computer network by offering the means to create and manipulate information containers with dynamic registers.

2. Description of the Related Art

In the present day, querying and usage of information resources on a computer network is accomplished by individuals directing a search effort by submitting key words or phrases to be compared to those key words or phrases contained in the content or description of that information resource, with indices and contents residing in a fixed location unchanging except by human input. Similarly, the class of storage medium upon which information resides, it class and subclass organizational structures, and its routes of access all remain fundamentally unaltered by ongoing user queries and usage. Only the direct and intended intervention of the owner of the information content or computer hosting site changes these parameters, normally accomplished manually by programmers or systems operators at their own discretion or the discretion of the site owner.

There exists currently in the art a limited means of interfacing a computer user with the information available on computer networks such as the world wide web. Primarily, these means are search engines. Search engines query thousands or tens of thousands of index pages per second to suggest the location of information while the user waits. While factual information can be accessed, the more complex, particular or subtle the inquiry, the more branches and sub-branches need to be explored in a time consuming fashion in order to have any chance of success. Further, there are no such automatic devices that reconstruct the information into more useful groupings or makes it more accessible according to factors attached to the content by the content creator such as the space or time relevancy of its content, or factors attached to the content by the system's compilation and analysis of the accumulated biography of that specific content's readership.

The utility of wide area and public computer networks is thus greatly limited by the static information model and infrastructure upon which those networks operate.

One problem is that on a wide area or public network, specific content such as a document remains inert, except by the direct intervention of users, and is modified neither by patterns or history of usage on the network, or the existence of other content on the network.

Another problem is that content does not reside in an information infrastructure conducive to reconstruction by expert rule-based, fuzzy logic, or artificial intelligence based systems. Neither the intelligence of other information users nor the expert intelligence of an observant network computer system can be utilized in constructing, or re-constructing information resources. Where content resides in a fixed location and structure, "information" becomes something defined by the mind of the information provider rather than the mind of the information user, where the actual construction and utility of information exists. Information remains, like raw ore, in an unrefined state.

Another problem is that the class of storage medium upon which data resides cannot be system or user managed and altered according to the actual recorded and analyzed hierarchically graded usage of any given information resource residing on that storage medium except by statistical analysis of universal, undefined "hits" or visits to that page or site.

Another problem is that information resource groupings remain fixed on the given storage medium location according to the original installation by the resource author, not altered according to the actual recorded and analyzed hierarchically graded usage of that given information resource. Content itself remains inert, with no possibility of evolution.

A further problem with the prior art is that neither the search templates generated by those more knowledgeable in a given field of inquiry, nor the search strategies historically determined to be successful, or system-constructed according to analyses of search strategies historically determined to be successful, are available to inquiring users. A search template is here defined as one or more text phrases, graphics, video or audio bits, alone or in any defined outline or relational format designed to accomplish an inquiry. Internet or wide area network search may return dozens of briefs to a keyword or key phrase inquiry sometimes requiring the

time-consuming examination of multiple information resources or locations, with no historical relation to the success of any given search strategy.

A further problem is that there is limited means to add to, subtract from, or alter the information content of documents, databases, or sites without communicating with the owners or operators of those information resources, e.g., contacting, obtaining permission, negotiating and manually altering, adding or subtracting content. Additionally, once so altered, there is not a means to derive a proportionate value, and thereby a proportionate royalty as the information is used.

A final problem is that the physical residence of a body of data or its cyberspace location may not serve its largest body of users in the most expedient manner of access. Neither the expert intelligence of other information users nor the expert intelligence of an observant computer system is presently utilized by inherent network intelligence to analyze, re-design and construct access routes to information medium except by statistical analysis of universal, undefined "hits" or visits to that page or site.

Therefore, there is a need for a system and methods for creating and manipulating information containers with dynamic interactive registers defining more comprehensive information about contained content in a computer, media or publishing network, in order to manufacture information on, upgrade the utility of, and develop intelligence in, a computer network by providing a searching user the means to utilize the searches of other users or the historically determined and compiled searches of the system, a means to containerize information with multiple registers governing the interaction of that container, a means to re-classify the storage medium and location of information resources resident on the network, a means to allow the reconstruction of content into more useful formations, and a means to reconstruct the access routes to that information.

SUMMARY OF THE INVENTION

The present invention is a system and methods for manufacturing information on, upgrading the utility of, and developing intelligence in, a computer or digital network, local, wide area, public, corporate, or digital-based, supported, or enhanced physical media form or public or published media, or other by offering the means to create and manipulate information containers with dynamic registers.

The system of the present invention comprises an input device, an output device, a processor, a memory unit, a data storage device, and a means of communicating with other computers, network of computers, or digital-based, supported or enhanced physical media forms or public or published media. These components are preferably coupled by a bus and configured for multi-media presentation, but may also be distributed throughout a network according to the requirements of highest and best use.

The memory unit advantageously includes an information container made interactive with dynamic registers, a container editor, a search interface, a search engine, a search engine editor, system-wide hierarchical container gateways interacting with dynamic container registers, a gateway editor, a register editor, a data collection means with editor, a data reporting means with editor, an analysis engine with editor, an executing engine with editor, databases, and a means of communicating with other computers as above. These components may reside in a distributed fashion in any configuration on multiple computer systems or networks.

The present invention advantageously provides a container editor for creating containers, containerizing storing information in containers and defining and altering container registers. A container is an interactive nestable logical domain configurable as both subset and superset, including a minimum set of attributes coded into dynamic interactive evolving registers, containing any information component, digital code, file, search string, set, database, network, event or process, and maintaining a unique network-wide lifelong identity.

The container editor allows the authoring user to create containers and encapsulate any information component in a container with registers, establishing a unique network lifelong identity, characteristics, and parameters and rules of interaction. The authoring user defines and sets the register with a starting counter and/or mathematical description by utilizing menus and simple graphing tools or other tools appropriate to that particular register. The registers determine the interaction of that container with other containers, system components, system gateways, events and processes on the computer network.

Containers and registers, upon creation, may be universal or class-specific. The editor provides the means to create system-defined registers as well as the means to create other registers. The editor enables the register values to be set by the user or by the system, in which case the register value may be fixed or alterable by the user upon creation. Register values are

evolving or non-evolving for the duration of the life of the container on the system. Evolving registers may change through time, space, interaction, system history and other means.

System-defined registers comprise: (1) an historical container register, logging the history of the interaction of that container with other containers, events and processes on the network, (2) an historical system register, logging the history of pertinent critical and processes on the network, (3) a point register accumulating points based upon a hierarchically rated history of usage, (4) an identity register maintaining a unique network wide identification and access location for a given container, (5) a brokerage register maintaining a record of ownership percentage and economic values, and others.

The present invention also includes user-defined registers. User defined registers may be created wholly by the user and assigned a starting value, or simply assigned value by the user when that register is pre-existent in the system or acquired from another user, and then appended to any information container, or detached from any container.

Exemplary user-defined registers comprise (1) a report register, setting trigger levels for report sequences, content determination and delivery target, (2) a triple time register, consisting of a range, map, graph, list, curve or other representation designating time relevance, actively, assigning the time characteristics by which that container will act upon another container or process, passively, assigning the time characteristics by which that container be acted upon by another container or process, and neutrally, assigning the time characteristics by which that container will interact with another container or process, (3) a triple space register, consisting of a range, map, graph, list, curve or other representation designating the domain and determinants of space relevance, actively, assigning the space characteristics by which that content will act upon another container or process, passively, assigning the space, characteristics by which that content will be acted upon by another container or process, and neutrally, assigning the space characteristics by which that container will interact with another container or process, (4) a domain of influence register, determining the set, class and range of containers upon which that container will act, (5) a domain of receptivity register, determining the set, class and range of containers allowed to act upon that container, (6) a domain of neutrality register, determining the set, class and range of containers with which that container will interact, (7) a domain of containment register, determining the set, class and range of containers which that container may logically encompass, (8) a domain of inclusion register, determining the set, class and range of containers by which that container might be encompassed, (9) an ownership register, recording the original ownership of that containers, (10) a proportionate ownership register, determining the proportionate ownership of that containers, (11) a creator profile register, describing the creator or creators of that container, (12) an ownership address register, maintaining the address of the creator or creators of that container, (13) a value register, assigning a monetary or credit value to that container, and (14) other registers created by users or the system.

Containers are nestable and configurable as both subset and superset and may be designated hierarchically according to inclusive range, such as image component, image, image file, image collection, image database, or if text, text fragment, sentence, paragraph, page, document, document collection, document, database, document library, or any arrangement wherein containers are defined as increasingly inclusive sets of sets of digital components.

The present invention also includes, structurally integrated into each container, or strategically placed within a network at container transit points, unique gateways, nestable in a hierarchical or set and class network scheme. Gateways gather and store container register information according to system-defined, system-generated, or user determined rules as containers exit and enter one another, governing how containers system processes or system components interact within the domain of that container, or after exiting and entering that container, and governing how containers, system components and system processes interact with that unique gateway, including how data collection and reporting is managed at that gateway. The gateways record the register information of internally nested sub and superset containers, transient containers and search templates, including the grade of access requested, and, acting as an agent of an analysis engine and execution engine, govern the traffic and interaction of those containers and searches with the information resource of which they are the gateway and other gateways. The gateways' record of internally nested and transient container registers, and its own interaction with those containers, is made available, according to a rules-based determination, to the process of the analysis engine by the data collection and/or data reporting means.

The present invention also includes a means of data storage at any given gateway.

The present invention also includes a data collection means, residing anywhere on the network, or located at one or more hierarchical levels of nestable container gateways for gathering information from other gateways and analysis engines according to system, system-generated or user determined rules. The data collection means manages the gathering of data regarding network-wide user choices, usage and information about information, by collecting it from container and gateway registers as those containers and gateways pass through one another. Such statistics as frequency, pattern, and range of time, space and logical class is collected as directed by the analysis engine, and made that data available to the analysis engine by advancing it directly to the analysis engine, or incrementally, to the next greater hierarchically inclusive collection level. The rules of data collection may be manually set or altered by the system manager, or set by the system and altered by the system in its evolutionary capacity.

The present invention also includes a data reporting means, located at one or more hierarchical levels of nestable container gateways for submitting information to other gateways and analysis engines according to system, system-generated or user determined rules. The data reporting means manages the sending of data from the registers, gateways and search templates in a frequency, pattern, and range of time, space and logical class as directed by the analysis engine, and makes that data available to the analysis engine by advancing it directly to the analysis engine, or incrementally to the next greater hierarchically inclusive reporting level. The rules of data collection may be manually set or altered by the system manager, or set by the system and altered by the system in its evolutionary capacity. The data reporting means may be established to work in concert, in redundancy, or in contiguous or interwoven threads of hierarchically nested containers.

The present invention also includes an analysis engine that receives, reports and collects information regarding the interaction of user searches with gateways and container registers, as well as container registers with other container registers, and container registers with gateways. The analysis engine analyzes the information submitted by the gateways and instructs the execution engine to create new information containers, content assemblages, storage schemes, access routes, search templates, and gateway instructions. The analysis engine includes an editor that provides a system manager with a means of editing the operating principles of that engine, governing data reporting, data collection, search template loading, gateway instructions, and other.

The present invention also includes an execution engine, fulfilling the instructions of the analysis engine, to create new information containers, content sun and superset assemblages, storage schemes, access routes, search templates, and gateway instructions. The execution engine includes an editor that provides a system manager with a means of editing the operating principles of that engine, governing data reporting, data collection, search template loading, gateway instructions, and other.

The present invention also includes a search interface or browser. The search interface provides a means for a searching user to submit, record and access search streams or phrases generated historically by himself, other users, or the system. Search streams or phrases of other users are those that have been historically determined by the system to have the highest probability of utility to the searching user. Search streams or phrases generated by the system are those that have been constructed by the system through the analysis engine based upon the same criteria.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a first and preferred embodiment of a system constructed according to the present invention.

FIG. **2 A** is block diagram of a preferred embodiment of the memory unit.

FIG. **2 B** is an exemplary embodiment of a computer network showing computer servers, personal computers, workstations, Internet, Wide Area Networks, Intranets in relationship with containers and gateways.

FIG. **2B1** is an exemplary embodiment of a computer network showing computer servers, personal computers, workstations, Internet, Wide Area Networks, Intranets in relationship with containers and gateways and exemplary locations of gateway storage in proximity to one or more of the various sites.

FIGS. **2C** through **2H** are exemplary embodiments in block diagram form of computer network components showing a possible placement of nested containers, computer servers, gateways, and the software components named in Fig. 2 A on a network.

FIG. **3A** is a graphical representation for one embodiment of a container having a plurality of containers nested within that container.

FIG. **3C** is a drawing showing elements that might be logically encapsulated by a container. FIG. **4** is a drawing of an information container showing a gateway and registers logically

encapsulating containerized elements.

FIG. **5** is a flowchart showing a preferred method for the containerization process and container editor operating on the communication device.

FIG. **6** is a flowchart showing a preferred method for searching for containers within a node.

FIG. **7** is a flowchart further showing a preferred method for searching for containers over one or more gateways.

FIG. **8** is a flowchart showing a method for performing the data collection and reporting on containers.

FIG. **9** is a flowchart showing the operation of the analysis engine.

FIG. **10** is a flowchart showing the operation of the execution engine.

FIG. **11** is a flowchart showing the operation of the gateway editor.

FIG. **12** is a flowchart showing the operation of the gateway process.

FIG. **13A** is a drawing showing an example of nested containers, gateways, registers, analysis engines and an execution engine prior to container reconstruction as depicted in 13 B, 13 C and 13 D.

FIG. **13B** is a drawing showing the reconstructed nested containers of Figure 13A.

FIG. **13C** is a drawing showing further reconstruction of nested containers, with a container relocated to reside within another container.

FIG. **13D** is a drawing showing a flowchart of the reconstruction process

FIG. **14** is a drawing showing the screen interface of the container editor.

FIG. **15** is a drawing showing the screen interface of the gateway editor.

FIG. **16** is a drawing showing the screen interface of the search interface.

FIG. **17** is a drawing of a generic application program showing a drop-down menu link, and a button link to the containerization process or container editor.

DESCRIPTION OF THE PREFERRED EMBODIMENT

THE SYSTEM

Referring now to FIG. **1**, a preferred embodiment of a system **10** for creating and manipulating information containers with dynamic interactive registers in a computer, media, or publishing network **201** in order to manufacture information on, upgrade the utility of, and develop intelligence in that network **201**, is shown. The system **10** preferably comprises an input device **24**, an output device **16**, a processor **18**, a memory unit **22**, a data storage device **20**, and a communication device **26** operating on a network **201**. The input device **24**, an output device **16**, a processor **18**, a memory unit **22**, a data storage device **20**, are preferably coupled together by a bus **12** in a von Neumann architecture. Those skilled in the art will realize that these components **24**, **16**, **18**, **22**, **20**, and **26** may be coupled together according to various other computer architectures including any physical distribution of components linked together by the communication device **26** without departing from the spirit or scope of the present invention, and may be infinitely nested or chained, both as computer systems within a network **202**, and as networks within networks **201**.

The output device **16** preferably comprises a computer monitor for displaying high-resolution graphics and speakers for outputting high fidelity audio signals. The output device **16** is used to display various user interfaces **110**, **125**, **210**, **300**, **510**, **610**, **710**, as will be described below, for searching for and containerizing information, and editing the container gateways, containers, container registers, the data reporting means and the data collection means, and the search, analysis and execution engines. The author uses the input device **24** to manipulate icons, text, charts or graphs, or to select objects or text, in the process of packaging, searching or editing in a conventional manner such as in the Macintosh of Windows operating systems.

The processor **18** preferably executes programmed instruction steps, generates commands, stores data and analyzes data configurations according to programmed instruction steps that are stored in the memory unit **22** and in the data storage device **20**. The processor **22** is preferably a microprocessor such as the Motorola 680(x)0, the Intel 80(x)86 or Pentium, Pentium II, and successors, or processors made by AMD, or Cyrix CPU of the any class.

The memory unit **22** is preferably a predetermined amount of dynamic random access memory, a read-only memory, or both. The memory unit **22** stores data, operating systems, and programmed instructions steps, and manages the operations of all hardware and software components in the system **10** and on the network **201**, utilizing the communication device **26** whenever necessary or expeditious to link multiple computer systems **202** within the network **201**.

The data storage device **20** is preferably a disk storage device for storing data and programmed instruction steps. In the exemplary embodiment, the data storage device **20** is a hard disk drive. Historical recordings of network usage are stored on distributed and centralized data storage devices **20**.

The preferred embodiment of the input device **24** comprises a keyboard, microphone, and mouse type controller. Data and commands to the system **10** are input through the input device **24**.

The present invention also includes a communication device **26**. The communication device **26** underlies and sustains the operations of, referring now also to Fig **2** the analysis **400** and execution **500** engines, the data reporting **600** and collection **700** means, the container editor **110**, the search interface **300**, and the search engine **320**, providing the means to search, access, move, copy, utilize or otherwise perform operations with and on data. The communication device **26** utilizes one or more of the following technologies: modem, infrared, microwave, laser, photons, electrons, wave phenomena, cellular carrier, satellite, laser, router hub, direct cabling, physical transport, radio, broadcast or cable TV or other to communicate with other computers, digital-supported television, computer networks, or digital-based or supported public or published media, or physical media forms, on any a local, wide area, public, or any computer-based computer supported, or computer interfaced network, including but not limited to the Internet. It also allows for the functioning and distribution of any container **100** or container component herein described to reside anywhere on any computer system in any configuration on that local, wide area, public, or corporate computer-based or computer related network, or digital-based or supported media form.

Referring now to Figure **2 A**, a preferred embodiment of the memory unit **22** is shown. The memory unit includes: an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510**, a search interface **300**, search engine **320**, analysis engine **400**, execution engine **500**, a data reporting module, **600**, a data reporting editor **610**, a data collection module **700**, a data collection editor **710**, screen interfaces (GUI's) **936**, menu or access buttons from generic computer programs **937**, and databases **900**, all residing in memory optimized between a data storage means **20** such as magnetic, optical, laser, or other fixed storage, and a memory means **22** such as RAM. The memory unit **22** functions by operating on communications network **12** with a communication device **26** on multiple computer systems **202** within the network **201**. These components will be described first briefly in the following paragraphs, then in more detail with reference to Figures 3 A through 17.

Those skilled in the art will realize that these components might also be stored in contiguous blocks of memory, and that software components or portions thereof may reside in the memory unit **22** or the data storage means **20.**

The present invention includes information containers **100** as noted above. The information container **100** is a logically defined data enclosure which encapsulates any element or digital segment (text, graphic, photograph, audio, video, or other), or set of digital segments, or referring now to FIG. **3 C,** any system component or process, or other containers or sets of containers. A container **100** at minimum includes in its construction a logically encapsulated portion of cyberspace, a register and a gateway. A container **100** at minimum encapsulates a single digital bit, a single natural number or the logical description of another container, and at maximum all defined cyberspace, existing, growing and to be discovered, including but not limited to all containers, defined and to be defined in cyberspace. A container **100** contains the code to enable it to interact with the components enumerated in 2 A, and to reconstruct itself internally and manage itself on the network **201.**

The container **100** also includes container registers **120**. Container registers **120** are interactive dynamic values appended to the logical enclosure of an information container **100,** and serve to govern the interaction of that container **100** with other containers **100**, container gateways **200** and the system **10**, and to record the historical interaction of that container **100** on the system **10**. Container registers **120** may be values alone or contain code to establish certain parameters in interaction with other containers **100** or gateways **200.**

The present invention also includes container gateways **200.** Container gateways **200** are logically defined gateways residing both on containers **100** and independently in the system **10**. Gateways **200** govern the interactions of containers **100** within their domain, and alter the registers **120** of transiting containers **100** upon ingress and egress.

The present invention also includes container gateway storage **205** to hold the data collected from registers **120** of transient containers **100** in order to make it available to the data collection means **700** and the data reporting means **600**, and to store the rules governing the operations of its particular gateway **200**, governing transiting containers upon ingress and egress, and governing the interactive behavior of containers **100** within the container **100** to which that gateway **200** is attached. Gateway storage **205** may be located on gateways **200** themselves, containers **100** or anywhere on the network **202, 201**, including but not limited to Internet, Intranet, LAN, WAN, according to best analysis and use.

The memory unit **22** also includes an execution engine **500** to perform the functions on the system **10** as directed by the analysis engine after its analysis of data from the data reporting means **600**, the data collection means **700**, and the search interface **300**.

The memory unit **22** also includes a search interface **300**, by which the user enters, selects or edits search phrases or digital strings to be used by the search engine **320** to locate containers **100**.

The memory unit **22** also includes an analysis engine **400** which performs rules based or other analysis upon the data collected from the search interface **300** and the data collection **700** and data reporting **600** means.

The memory unit **22** also includes a data reporting means **600**, by which means the information collected by gateways **200** from transient containers **100** is sent to the analysis engine **400**.

The memory unit **22** also includes a data collection means **700**, by which means the analysis engine **400** gathers the information collected by gateways **200** from transient containers **100**.

The memory unit **22** also includes a container editor **110** for creating, selecting, acquiring, modifying and appending registers **120** and gateways **200** to containers **100**, for creating, selecting, acquiring, and modifying containers, and for selecting content **01** to encapsulate.

The memory unit **22** also includes a register editor **125**, for creating, selecting, acquiring and modifying container registers **120** and establishing and adjusting the values therein.

The memory unit **22** also includes a gateway editor **210**, by which means the user determines the rules governing the interaction of a given gateway **210** with the registers **120** of transient containers **100,** governing transiting containers upon ingress and egress, and governing the interactive behavior of containers within the container to which that gateway is attached.

The memory unit **22** also includes databases **900,** by which means the analysis engine **400,** the execution engine **500**, the gateways **100**, the editors **110**, **125**, **210**, **510**, **610**, **710**, and the search interface **300**, store information for later use.

The memory unit **22** present invention also includes a search engine **320** by which means the user is able to locate containers **100** and, referring now to Fig. **4**, containerized elements **01**.

The memory unit **22** present invention also includes an engine editor **510,** by which means the user establishes the rules and operating procedures for the analysis engine **400** and the execution engine **500.**

The memory unit **22** present invention also includes a reporting means editor **610**, by which means the user establishes the rules and schedule under which the information collected by gateways **200** from transient containers **100** will be sent to the analysis engine **400.**

The memory unit **22** present invention also includes a collection means editor **710,** by which means the user establishes the rules and schedule under which the analysis engine **400** will gathers the information collected by gateways **200** from transient containers **100.**

The memory unit **22** present invention also includes screen interfaces (GUI's) **936**, specifically designed to simplify and enhance the operations of the container editor **110**, the gateway editor **210**, and the search interface **300.**

The present invention also includes a menu or button access **937**, by which a user utilizing any generic computer program may access the system **10** or the container editor **110** from a menu selection(s) or button(s) within that program.

The present invention also includes a computer, media or publishing network **201**, comprising computers, digital devices and digital media **202** and a communication device **26**, within which the components enumerated in Fig. **2 A** interact, compiling, analyzing, and altering containers **100** and the network **201**according to information gathered from container registers **120.**

The memory unit **22** also includes one or more computers **202**, by which means the components of Fig **1** sustain the operations described in Fig. **2 A.**

The memory unit **22** also includes flat or relational databases **900**, used where, and as required. Databases are used to store search phrases, search templates, system history for the analysis engine and execution engine, container levels and container, sites and digital elements, or any and all storage required to operate the system.

Referring now to FIG. **2 B,** a drawing of a computer network **201** as a system **10,** showing a possible placement of nested containers **100**, computer servers, gateways **200**, on the sites described below. (Note: Fig. 2 B utilizes in parts the same numbering scheme as Fig. 13 A, 13 B, 13 C, 13 D and as Fig. 2 A.) In FIG. **2 B** various exemplary sites are shown, any or all of which might interact dynamically within the system. Site 1 shows a single workstation with a container and gateway connected to an Intranet. .(Individual containers may be a floppy or CD-Rom to be downloaded or inserted.) Site 2 shows a server with a gateway in relationship to various containers.. Site 3 shows an Internet web page with a container residing on it. Site 4 shows a personal computer with containers and a gateway connected to the Internet. Site 5 shows a configuration of multiple servers and containers on a Wide Area Network.. Site 6 shows a workstations with a gateway and containers within a container connected to a Wide Area Network. Site 7 shows an independent gateway, capable of acting as a data collection and data reporting site as it gathers data from the registers of transiting containers, and as an agent of the execution engine as it alters the registers of transient containers. A container **100** contains the code to enable it to interact with the components enumerated in **2A,** and to reconstruct itself internally and manage itself on the network **201.** The code resides in and with the container in its registers and gateway definitions and controls. Additional system code resides in all sites to manage the individual and collective operation and oversight of the components enumerated in **2A**, with the specific components distributed amongst the sites according to the requirements of optimization.

Referring now to Fig. **2 B 1** various exemplary sites are shown as described above in Fig. **2 B**, with the addition of possible location of one or more gateway storage **205** locations.

Referring now to Figures **2 C** through **2 H**, various exemplary sites with one or more of the logical components of the system 10 in relationship are shown. Site 1 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510**, a search interface **300**, search engine **320**, analysis engine **400**, execution engine **500**, a data reporting means **600**, a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 2 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510,** search engine **320**, analysis engine **400**, execution engine **500**, a data reporting means **600**, a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20,** utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 3 comprises an interactive information container 100, a container editor **110**, container registers **120**, a container register editor **125**, hierarchical container gateways 200, gateway storage 205, gateway editors **210**, and databases **900**, all residing on data storage means **20,** utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 4 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, a search interface **300**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Site 5 comprises an interactive information container **100**, container registers **120**, a container register editor **125**, hierarchical container gateways **200**, gateway storage **205,** and databases **900**, all residing on data storage means **20**, accessed and utilized by non-resident memory unit **22**, operating on communications network **12** with a communication device **26.**

Site 6 includes an independent analysis engine **400,** execution engine **500**, data collection means **700**, and data reporting means **600** gateway editors **210**, engine editors **510,** a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Referring now to FIG. **3 A** and FIG. **3 B**, a block diagram of several nested information containers is shown, including examples of elements, e.g., code **1100**, text **1200**, audio **1300**, video **1400**, photograph **1500**, graphic images **1600**, and examples of possible container level classifications in increasing size, e.g., element **10900000**, document **10800000**, database **10700000**, warehouse **10600000**, domain **10500000**,and continuing increasingly larger on Fig **3 (B)**, subject **10400000**, field **10300000**, master field **10200000**, species **10100000**. Containers may be infinitely nested and assigned any class, super class or sub class scheme and description by the creator of the container to govern nesting within that container. In addition to digital elements, containers may also include system process and components, including containerization itself.

Referring now to FIG. **3 C**, a block diagram of an information container system is shown, listing, without any relationship indicated, some of the possible system components and processes, or sets thereof, that may be encapsulated as elements **01** in an information container **100.** An information container 100 may include one or more of the following: any unique, container **100**, gateway **200**, output device **16**, input device **24**, output device process **160**, input device process **240**, data storage device **20**, data storage device process **2000**, processor **18**, bus **12**, content **01**, search process **02**, interface **04**, memory unit **22**, communication device **26**, search interface **300**, search process **98**, network **201**, class of device, process or content **999**, class of process at any unique class of device **990**, process at any unique device **99**, editor **110**, **125**, **210**, **510**, **610**, **710**, engine **320**, **400**, **500**, containerization process **1098**, or process **08**.

Any container may include (n) other containers, to infinity. The use of value evolving container registers **120** in conjunction with gateways **200**, data reporting modules **600**, data collection modules **700**, the analysis engine **400**, and the execution engine **500** provides the information container **100** with extensive knowledge of the use, operation of its internal contents, prior to, during and after those contents' residence within that container **100**, and extensive knowledge of the use, operation and contents of the system **10** external to itself, and allows the container **100** to establish and evolve its own identity and course of interaction on the system **10**. Further, containers **100**, as logical enclosures, can exist and operate independent of their digital contents, whether encapsulating audio, video, text, graphic, or other.

Referring now to FIG. 4, a block diagram of an information container **100** is shown. The information container **100** is a logically defined data enclosure which encapsulates any element, digital segment (text, graphic, photograph, audio, video, or other), set of digital segments as described above with reference to FIG. **3 (C)**, any system component or process, or other containers or sets of containers. The container **100** comprises the containerized elements **01**, registers **120** and a gateway **200**.

Registers **120** appended to an information container **110** are unique in that they operate independently of the encapsulated contents, providing rules of interaction, history of interaction, identity and interactive life to that container **100** through the duration of its existence on a network **201**, without requiring reference to, or interaction with, its specific contents. They enable a container **100** to establish an identity independent of its contents. Additionally, registers **120** are unique in that their internal values evolve through interaction with other containers **100**, gateways **200**, the analysis engine **400**, the execution engine **500**, and the choices made by the users in the search interface **300**, the container editor **110**, the register editor **125**, the gateway editor **210**, the engine editor **510**. Registers **120** are also unique in that they can interact with any register of a similar definition on any container **100** residing on the network **201**, independent of that container's contents. Registers **120**, once constructed, may be copied and appended to other containers **100** with their internal values reset, to form new containers. Register values, when collected at gateways **200** and made available to the analysis engine **400** through the data collection means **700** and the data reporting means **600**, provide an entirely new layer of network observation and analysis and operational control through the execution engine **500**. Registers **120** accomplish not only a real time information about information system, but also a real time information about information usage on a network. Further, because the user base of a network determines usage, the system **10**, in gathering information about information usage, is observing the choices of the human mind. When these choices are submitted to the analysis of a rules-based or other analysis engine **400**, the system **10** becomes capable of becoming progressively more responsive to the need of the user base, in effect, learning to become more useful by utilizing the execution engine **500** to create system-wide changes by altering the rules of gateway **200** interaction and thereby altering the registers **120** of transient containers **100** and establishing a complete evolutionary cycle of enhanced utility.

Further, in establishing the pre-defined registers as described in the following four paragraphs, the following unique aspects of information about information are utilized for the first time: 1) the dynamic governance of information according to its utility through time, in active, passive and neutral aspects, as explained below; 2) the dynamic governance of information according to its utility through space in active, passive and neutral aspects, as explained below; 3) the dynamic governance of information according to its ownership, as explained below; 4) the dynamic governance of information according to its unique history of interaction as an identity on a network, as explained below; 5) the dynamic governance of information according to the history of the system on which it exists, as explained below; 6) the dynamic governance of information according to established rules of interaction, in active, passive and neutral aspects, as explained below; 7) the dynamic governance of information according to the profile of its creator, as explained below; 8) the dynamic governance of information according to the value established by its ongoing usage, as explained below; 9) the dynamic governance of information according to it distributed ownership, as explained below; 10) the dynamic governance of information according to what class of information it might be incorporated into, and according to what class of information container it might incorporate, as explained below; 11) the dynamic governance of information according to self-reporting, as explained below.

Referring now to Fig **4**, registers **120** may be (1) pre-defined, (2) created by the user or acquired by the user, or (3) system-defined or system-created. Pre-defined registers **120** are those immediately available for selection by the user within a given container editor as part of that container editor, in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. Registers **120** created by the user are those conceived and created by a specific user or user group and made immediately available for selection by the user or user group in conjunction with any of a wide number of container editors, in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. Registers **120** acquired by the user are those registers existing network-wide **201**, created by the user base, that might be located and acquired by the user in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. System-defined registers are those registers whose values are set and/or controlled by the system **10**. System-created registers are those registers created by the system **10**.

Registers **120** are user or user-base created or system-created values or ranges made available by the system **10** to attach to a unique container, and hold system-set, user-set, or system-evolved values. Values may be numeric, may describe domains of time or space, or may provide information about the container **100**, the user, or the system **10**. Registers **120** may be active, passive or interactive and may evolve with system use. Pre-defined registers include, but are not limited to, system history **110000**, container history **101000**, active time **102000**, passive time **103000**, neutral time **104000**, active space **111000**, passive space **112000**, neutral space **113000**, containment **105000**, inclusion **106000**, identity **114000**, value **115000**, ownership **107000**, ownership addresses **116000**, proportionate ownership **117000**, creator profile **108000**, receptivity **118000**, influence **119000**, points **109000**, others **120000**, reporting **121000**, neutrality **122000**, acquire **123000**, create **124000**, content title **125000**, content key phrase(s) **126000**, and content description **127000**, security **12800**, and parent rules **129000**.

Pre-defined registers comprise an historical container register **101000**, logging the history of the interaction of that container **100** with other containers, events and processes on the network **201**, an historical system register **110000**, logging the history of pertinent critical and processes on the network, a point register **109000** accumulating points based upon a hierarchically rated history of usage, an identity register **114000** maintaining a unique network wide identification and access location for a given container specifying a unique time and place of origin and original residence, a proportionate ownership register **117000** maintaining a record of ownership percentage and economic values, and others **120000**.

User-defined registers include a report register **121000** setting trigger levels for report sequences, content determination and delivery target, three time registers, consisting of a range, map, graph, list, curve or other designating time relevance, **102000** assigning the time characteristics by which that container will act upon another container or process, **103000** assigning the time characteristics by which that container be acted upon by another container or process, and **104000** assigning the time characteristics by which that container will interact with another container or process, three space registers, consisting of a range, map, graph, list, curve or other designating the domain and determinants of space relevance, **111000** assigning the space characteristics by which that content will act upon another container or process, **112000** assigning the space, characteristics by which that content will be acted upon by another container or process, and **113000** assigning the space characteristics by which that container will interact with another container or process, a domain of influence register **119000**, determining the set, class and range of containers upon which that container will act, a domain of receptivity register **118000**, determining the set, class and range of containers allowed to act upon that container, a domain of neutrality register **122000**, determining the set, class and range of containers with which that container will interact, a domain of containment register **105000**, determining the set, class and range of containers which that container may logically encompass, a domain of inclusion **106000** register, determining the set, class and range of containers by which that container might be encapsulated, an ownership register **107000**, recording the original ownership of that containers, a creator profile register **108000**, describing the creator or creators of that container, an ownership address register **116000**, maintaining the address of the creator or creators of that container, a value register **115000**, assigning a monetary or credit value to that container, other registers **120000** created by users or the system, a reporting register **121000**, determining the content, scheduling and recipients of information about that container, a neutrality register **122000**, an acquire register **123000**, enabling the user to search and utilize other registers residing on the network, a create register **124000**, enabling the user to construct a new register, a content title register **125000**, naming the contents of the container, a content key register, **126000**, identifying the container contents with a key phrase generated by the user and/or the system based upon successful usage of that phrase in conjunction with the utilization of the information within that container **100**, a content description register **127000**, identifying the container contents with additional description, a security register **128000**, controlling container security, and a parent container register **129000**, storing the rules governing container interaction as dictated by the parent (encapsulating) container.

The container also includes a gateway **200** and gateway storage **205.**

Gateways **200** are logically defined passageways residing both on containers **100** and independently in the system **10**. Gateways **200** govern the interactions of containers **100** encapsulated within their domain by reading and storing register **120** information of containers entering and exiting that container **100**.

The present invention also includes container gateway storage **205.** Gateway storage **205** stores information regarding the residence, absence, transience, and alteration of encapsulated and encapsulating containers **100**, and their attached registers **120, holding** the data collected from registers **120** of transient containers **100** in order to make it available to the data collection means **700** and the data reporting means **600**, and storing the rules governing the operations of its particular gateway **200**.

Referring now to FIG. **5**, a flow chart of the preferred method for creating a container **100**.is shown.

Input is received from the user selecting a container level through use of a drop-down menu **10100**. A menu of all possible container classes within the subset and superset scheme of multiple hierarchically nested containers, i.e.; element, document, file, database, warehouse, domain, and more, is displayed on the output device **10200**. Input is received from the user selecting a class **10300**.

A graphic representation of a container in that class, with registers common to all containers as well as registers unique to its class is displayed **10301**.

Input is received from the user choosing to "create" **10400**, "edit" **10500**, or "locate" **10600**.

When the input of "create" **10400** is received from the user, a container template in that class appears **10410**. Input from the user is then received adding or selecting a register 10540 to append to that container template. When input is received from the user adding a register, a list of registers that might be added to that class of container is made available to select **10550**. Input is received from the user selecting a register **10560** and editing it **10570**. The menu returns to "add or select" **10540**.

If the input of "locate" **10600** is received from the user, the system prompts the user to enter the identity of the container or class of containers **10605**. The system locates the container(s) **10610**. Input is received from the user selecting a container **10620**. The system prompts the user for a security code for permission to access the container for template use, or to alter its registers, or to alter its content **10630**. . Input is received from the user entering a name and password providing access to one of the security levels **10640.** Input is received from the user editing the container accordingly by transition to step **10500** and performing the steps for editing.

If the input of "edit" **10500** is received, a list of containers available to edit at that level is shown **10510**. Input is received from the user selecting a container **10520**. That container appears, available to edit **10530**. Input is received from the user selecting "add" or "select" registers **10540** by the user clicking on the graphically depicted register, or from a drop down menu. Input is received from the user selecting the register to edit **10560**. Input is received from the user selecting "modify" or "delete" for that register **10565**. If input is received from the user to “delete,” that register is severed from the container. If input is received from the user to “modify”, the register editor **10570** screen appropriate to that register appears, i.e., an x-y type graph to define a curve of relevant active time, in which the user manipulates the x-y termini, scale and curve, or a global map in which Input is received from the user selecting the locale of active space, whether zip code, city, county, state, country, continent, plant or other. When input is received from the user saving the definition, the screen returns to the main container screen to make another selection available. . Input is received from the user defining as many registers as he chooses. One of the registers may be named "new register.” Input is received from the user selecting the new register, and if chosen by the user, defining a wholly unique and new kind of register by the user entering input into the register editor **125.**

When the input is received from the user choosing to add a register, a list of registers that might be added to that class of container are made available to select **10550**. Input is received from the user selecting a register **10560** and editing it **10570**. The menu returns to “add or select” **10540**, and in turn to Input – Select Container.

Input may then be received from the user choosing to add, modify, or delete the container contents **10700**. Once the registers are defined, input is received from the user indicating completion and the interface reverts to the container editor. When input is received from the user choosing "select component" (to select the component to containerize) from the main menu bar **10700, a** window appears allowing the user to select any file, component, or other container. If for example, the user were creating a warehouse container, and wishes to incorporate several databases into that container, input would then be received from the user selecting "database.” The program would prompt the user for the location (directory) of that database or container. If the requested selection is not containerized, input may then be received from the user choosing to containerize the element at that time, after which the program returns to "select component." Once input is received from the user defining the database location, the program logically encases the directory or directories in the defined container. The above procedure may be repeated as many times as desired to include multiple databases within a single container. While logical simplicity would dictate that all containers within a container be of the same subset, it would be possible for input to be received from the user choosing containers of any subset to include in the container. When input is received from the user choosing "finished,” the container is created with a unique network identity, preferably through some combination of exact time and digital device serial number, or centralized numbering system, or other means. The container **100** contains all digital code, including data and program software from the selected items or containers.

Input may then be received from the user to publish the container **11100** at a user-identified or system suggested location **11200** to be selected **11400**.

Input is received from the user to "publish", from the main menu bar **11100**. Input is received from the user choosing to leave the container where it was created, move or copy it to another drive, directory, computer, or network the user designates, or select the location from location options offered by the system **11200**, or submit, or duplicate and submit, the container to the analysis engine **400** for intelligent inclusion in other containers, thus allowing the system to publish the container as instructed or choose the residence of the container **11400**.

If input is received from the user to choosing to "move," or "copy" a browse function allows the user to name the new location or browse a list of possible locations. If input is received from the user choosing to "submit," a browser function allows the user to name the analysis search engine **310** or browse a list of possible analyses engines. When input is received from the user choosing the residence of the container **11300**, the program restores the search interface screen.

Referring now to FIG. **6**, a flow chart of the method for searching for containers **100**.

When input is received from the user selecting "search interface" from the main title bar, the search interface screen appears. The user is given the choice of containerizing selected content or requesting that container levels be displayed **30100**. From a drop down menu another menu appears allowing input to be received from the user selecting the container level **30200**. Input is received from the user selecting the container level (from the smallest component to the whole system) **30300.**

Input is received **30310** from the user selecting the phrases, containers or components, which then are re-submitted to the same process, until the input is received from the user selecting a specific site or container.

The search phrase, whether containerized or not, is submitted simultaneously to the search engine **30400** and the analysis engine **30500**.

The screen then reports in a selection menu, the number of applicable sites found by the search engine **30410**, the number of historically proven applicable sites found by the analysis engine **30410**, the number of historically proven applicable containers at the selected container level or any container level found by the analysis engine **30410**, and the number of historically proven new search phrases or digital segments found by the analysis engine **30320**. . Input is received from the user selecting one of the named sets above **30330**. If input is received from the user choosing the search engine, the search interface lists the applicable site titles with a brief description **30410**. If input is received from the user choosing the site list of the analysis, the search interface lists the applicable site titles with a brief description **30410**. . If input is received from the user choosing the container list of the analysis engine, the search interface lists the applicable container titles with a brief description **30410**. . If input is received from the user selecting a container **30420**, the system offers the means to view titles and descriptions of sub-containers at any chosen class level. . If input is received from the user choosing the phrase list of the analysis engine, the search interface lists the applicable phrases or digital segments with a brief description **30320**. The search and search result cycle repeats until input is received from the user choosing to go to an individual container or site.

Input is received from the user entering text or any digital string describing his search objectives into a text or search box. When input is received from the user submitting the search string, the system provides the option of containerizing the search through the container editor **110.** Once the search container **101** is created, the system restores the search interface **300** screen the user.

Input is received from the user selecting "search", "supported search" or "both" from another drop-down menu and from submitting the search. When input is received from the user selecting "search" 30310, the search phrase is submitted to the search engine **30400**, which searches both content and the appropriate container registers, as pre-indexed in the search engine, and returns a list of appropriate locations, components or containers. When input is received from the selecting "supported search", the search phrase is submitted to the analysis engine search support, which returns a list, in a drop-down menu, of search phrases or individual containers, for any and all container levels, used by other users or created by the system and known to be historically successful for the described effort and the described searching user, as per the results of the analysis search engine. Input is received from the user selecting a new search phrase or specific container from the drop down menu **30330**. When input is received from the user choosing a new search phrase, that phrase is also submitted to the analysis engine **30500** which returns a list of pre-compiled historically proven sites, components or containers associated with that search phrase **30320**. Input is received from the user choosing a selection **30420** and the system calls up that specific site, container or component. If input is received from the user selecting a specific site, container or component at any time during the search process, that element is called up by the system **30440**.

Input is received from the user choosing to containerize a search or select a container level in which to search **30100**. When input is received from the user choosing to containerize the search, the software moves to the container editor as described in Fig. **5**, and then returns the user to the search interface screen. Input is received from the user selecting to search a specific container level or the whole network. The system shows the available levels **30200**. Input is received from the user selecting a container level **30300**, and entering the text or digital component comprising the search string **30310**. The system searches the containers **30400** while simultaneously submitting the search string to the analysis engine **30500**. While the system is accessing containers, sites or templates **30700**, the analysis engine **30500** inquires of the appropriate database **30600** to access historically successful containers, sites or search templates corresponding to the search request **30700**, which is then shown on another portion or option of the search interface, either as available containers or sites **30410** or as search template options **30320**. On one portion or option of the search interface screen the corresponding containers or sites are listed and/or previewed for selection **30410**. Input is received from the user selecting the container to access **30420**. The system accesses that container **30430** and shows it on the screen **30440** for user review. Input is received from the user selecting an operation, i.e., preview, read, purchase, move, copy, lease, in any composed schedule with operations assigned specific values **30460**, and the system obtains the specified result **30470**. The selection of the operation including any interaction with any uniquely defined container **100** is recorded **30800** by the container gateway (Fig. 2 A, **200**), stored in the gateway storage **205** and made available to the analysis engine (Fig. **9**) by the data collection and reporting means (Fig. **8**). Reporting and collection occurs on a regular basis according to user determined times or rules. The analysis engine compiles and analyzes selections according to various rules-based systems applicable to the particular container area of residence in cyberspace.

. Input is received from the user selecting the container or site **30410**, proceeding as described above, or selecting a search template **30330**, and editing it to re-enter the search

30310. All operations on Fig. **6** utilize the communication device **26** whenever necessary or expeditious.

Referring now to FIG. **7**, a flow chart of the search process is shown. Steps in FIG. **7** repeated from FIG. **6** are given the same reference number as in FIG. **6** for convenience and ease of understanding. Fig. **7** commences with "SEARCH TRANSITS GATEWAY **32100**", continuing from Fig. **6**, "SYSTEM SEARCHES CONTAINERS **30400**". The submitted search **32100** transits the gateway **200**. The gateway **200** interacts with the container registers **32200**. The gateways **200** store the information downloaded from the registers **32300**, and the container registers are altered **32500**. The container registers **120** then interact with the registers **120** of the encapsulated search, which registers, and the values set within, have been constructed and appended to the search through the search interface **32600.** Values are exchanged and compared and operations performed under the rules governing both interacting containers **100**, and the rules governing the search container **100** and any gateway **200.** The search engine **320,** operating under the principles and means of search engines presently existing as described elsewhere, **then** provides to the search interface **32600** a list of containers **100** meeting the requirements of the search and its appended registers, as well as additional search options **32900**. The gateway **200** reports and makes available for collection to the analysis engine **400** the information obtained from the interaction **32400**. On a periodic basis defined by the user or a rules-based system, the analysis engine **400** (Fig. **9**) stores in databases **900**, analyzes and instructs the execution engine **500**, and the execution engine **500** executes changes in the system components as defined below. (Fig. **10**). All operations on Fig. **7** utilize the communication device **26** whenever necessary or expeditious.

On the remaining figures, shapes referring to other figures, to operations external to the scope of the present figures, or to the subject of the present drawing, are indicated with dashed lines, and are shown only to place the described operations in the context of continuous and continual operations external to the drawing.

Referring now to FIG. **8**, a flow chart of the preferred process for collecting and reporting information on containers is shown. The data reporting **600** and data collection **700** means utilizes subroutines within the analysis engines **400** and gateways **200** to submit and collect register information and sub level analysis to other analysis engines **400** or other gateways **200** of a higher (larger) logical set in a set pattern and frequency defined by the administrator.

Input is received from the user selecting "data reporting" **70100** from the "edit gateway" drop-down menu. Container levels are displayed **70200**. Input is received from the user selecting container level **70300.** A menu of all possible gateways **70320** and analysis engines **70330** residing on gateways on the above defined container class appears, depicted graphically as a tree of analysis engines and gateways at that container level. Input is received from the user selecting "source" from "source or destination." Input is received from the user **70400** selecting a container, containers, or class of container by clicking on the graphically depicted container(s) or container level on a display device. Input is received from the user **70410** selecting "destination" from "source or destination" Input is received from the user **70500** selecting an analysis engine, analysis engines, or class of analysis engine by clicking on the graphically depicted analysis engine(s) or analysis engine level on a display device. A time scheduler is displayed. Input is received from the user **70510** selecting the reporting frequency for the selected gateways to report data to the selected engines. The data from the gateways is thenceforth continuously moved or copied to the analysis engines by the system **10** utilizing the execution engine **500** according to the defined schedule, rules and pattern **70420, 70520.**

Input is received from the user selecting "choose container level" **70300** from the gateway editor drop-down menu. A menu **70320** appears listing the classes of containers on the system within the defined subset and superset scheme of multiple hierarchically nested containers, i.e.; element, document, file, database, warehouse, domain, appears. Input is received from the user selecting the class of containers. A graphic representation of that container level throughout the system appears. Input **70300** is received from the user selecting individual containers or all the containers in that class.

From the gateway editor drop-down menu input **70100** is received from the user selecting "data collecting" A menu of all possible gateways and analysis engines residing on gateways on the above defined container class appears, depicted graphically as a tree of analysis engines, and gateways at that container level. Input **70510** is received from the user selecting "source" from "source or destination." Input is received from the user selecting a container, containers, or class of container by clicking on the graphically depicted container(s) or container level. Input **70510** is received from the user selecting "destination" from "source or destination." Input **70510** is received from the user selecting an analysis engine, analysis engines, or class of analysis engine by clicking on the graphically depicted analysis engine(s) or analysis engine level. A time scheduler appears. Input **70510** is received from the user selecting the collecting frequency for the selected engines to collect data from the selected gateways. The data from the gateways is thenceforth continuously moved or copied to the analysis engines by the system **10** utilizing the execution engine **500** according to the defined schedule, rules and pattern.

The data collection **700** means, utilizing the communication device **26** and an execution engine **500,** comprises one or more subroutines or agents programmed to travel through the network collecting the accumulated data and analyses from selected analysis engines, gateways or selected subset level of analysis engines or gateways (as above) in a pattern and frequency defined by the gateway administrator at a given container level. Input **70510** is received from the user or administrator, defining the collection and reporting of data, thus controlling permission within his gateway, and being subject to permission levels defined by others beyond his gateway.

Input is received from the user or gateway administrator selecting collection or reporting **70100** and the system shows the container levels available **70200**. Input is received from the user selecting a container level **70300**. Input is received from the user selecting "gateway" **70400** or "engine" **70500**. The system shows gateways **70320** or engines **70330** associated with that level. Input is received from the user editing the reporting parameters associated with a gateway or a class of gateways **70410** or an engine or class of engines **70510**. Input is received from the user selecting the collecting frequency for the chosen engines. When input is received from the user choosing to user save the definition, the screen returns to the main container screen, step **70100** to make another selection available. Input is received from the user choosing to repeat the cycle, choosing "destination" to describe the destination analysis engines and the data collecting frequency from those destination analysis engines. The data collection means **700** collects the accumulated gateway information in a pattern and frequency defined by the gateway administrator or user at a given container level.

The system utilizing the execution engine (see Fig. **10**) distributes the new parameters to the gateways **70420** or engines **70520** by the communication device **26**. Using the new parameters the gateways report to the analysis engines **70430** after, in some cases, conducting sub-analysis **70440,** or using sub-analysis **70440** to submit directly to specified gateways under certain conditions and parameters, and the analysis engines collect from the gateways **70530**.

The analysis engine uploads, downloads and utilizes information to databases **900** to conducts its analysis.

The invention includes an analysis engine **400**. Through the data reporting **600** means and data collection **700** the analysis engine **400** receives data and sub-analysis from the search interface and the gateways. Data includes, for each gateway **200**, the frequency and grade of access, the description of the user accessing, the identity of the container **100** accessing, the register parameters, and the historically accumulated register data.

Referring now to FIG. **9**, a flow chart of the operation of the analysis engine **400** is shown. Analysis engines **400** may reside at any gateway or anywhere in the system 10. The analysis engine **400**, operating under its own programmed sequence, utilizing the communication device **26**, works, by means of programmed rules of logical, mathematical, statistical or other analysis upon gateway and register information, in continuous interaction with the search process **410** and the data collection and reporting process **420** to analyze, determine and compile instructions **40100** on container construction **40110** to containerize in an automated process **40115**, on container contents **40120** to move, copy or delete containers **40125**, on storage schemes **40130** to move or copy containers to new storage **40135**, on access routes **40140** to alter gateway pointers to sought information **40145**, on search templates **40150** to add, delete or change search phrases and the referenced objects indicated by those search phrases **40155** and on gateway instructions **40160** to alter gateway registers and pointers **40165**.

Thus, analyses might include, but are not limited to, the physical locus of the users accessing, the demographic classification of the users accessing, the access frequency for a given container, the range or curve of time relevance affecting a container, the range or region of space relevance affecting a container **100**, the number or number of a specific type of container **100** transiting a gateway **200**, the hierarchically graded usage of containers **100** or container contents **01** compared with the demographic of those users accessing the container, the hierarchically graded usage of containers **100** or container contents **01** compared with search phrases entered into the search interface **300**, the hierarchically graded usage of containers **100** or container contents **01** compared with search phrases entered into the search interface 300 compared with the demographic of the users accessing, the number of pertinent containers nested within a given container **100**. Once an analysis is accomplished, the result is compared to pre-programmed rules triggering instruction sets (such as moving a container to nest within another container).

Instructions are then sent to the execution engine **40200**, which utilizes the communication device **26** to execute the instructions derived from the analyses. These containerized instructions transit the gateways **40300** and are utilized in the gateway process (Fig. **12**)

Referring now to FIG. **10**, a flow chart of the operation of the execution engine is shown. The execution engine **400**, operating under its own programmed sequence in response to the instructions from the analysis engine **50100**, utilizing the communication device **26**, works in continuous process as its containerized execution instructions transit the gateways **50200** to create containers **50210** in an automated containerization process **50215**, alter container contents **50230** by moving or copying containers to new containers **50235**, to alter storage **50240** by moving or copying containers to new storage **50245**, to alter access routes **50250** by altering gateway pointers **50255**, to alter search templates **50260** by adding, changing and deleting search phrases and the referenced objects indicated by those search phrases **50265**, to alter gateway instructions **50270** by altering gateway registers and pointers **50275**. The execution works in a continuous loop with the gateway process **50300**, the data collection and reporting process **50400** and the analysis engine process **50300**.

The invention includes gateways **200**. Gateways may be placed and reside anywhere on the network where containers transit. Gateways also reside on any or all containers. The gateway reads and stores the chosen register information from transient containers entering or exiting its logical boundaries. The resident analysis search engine, if any, performs the specified level of analysis. Data and analysis is both held for the collection means according to the pattern and timing specified in the data reporting **600** editor and submitted according to the pattern and timing specified in the data collection means editor **700**.

The gateways are network-wide, hierarchical, and nestable, and reside with a container encompassing any component, digital code, file, search string, set, database, network, event or process and maintaining a unique lifelong network wide identity and unique in all the universe historical identity, or may be strategically placed at such container transit points to gather and store register information attached to any such container, according to system-defined, system-generated, or user determined rules residing in its registers defining the behavior of those containers and components as they exit and enter one another, or interact with one another or any system process or system component within the logical domain of that container, or after exiting and entering that container, or defining how they interact with that unique gateway.

Gateway's registers comprise both system-defined and user-defined registers, alterable by author, duration, location, network-wide history, individual container history and/or interaction with other containers, gateways, networks or media, and evolve according to that gateway's history on a computer network, or according to the network history of events and processes, or according to that information component's interaction with other information containers, components, system components, network events or processes.

Referring now to FIG. **11,** a flow chart of the gateway editor is shown. From the main title bar input is received from the user selecting "containerize" or "gateway level" **20100.** When input is received from the user selecting "containerize" the system enters the container editor process **110.** When input is received from the user selecting "gateway," the system shows the gateway levels available **20200.** A menu of all possible gateways within the subset and superset scheme of defined multiple hierarchically nested gateways appears. Input is received from the user selecting the gateway level **20300.** The system searches the gateways **20500** to locate the available gateway templates **20700** and the available gateways **20600.** Input is received from the user selecting the gateway **20610** or gateway level template **20720.** The system goes to the gateway **20620** or to the template **20720.** A graphic representation of the chosen gateway **20630** or template **20730** appears. Input is received from the user to edit **20640** or create a gateway **20740.** Once completed, input may be received from the user selecting "analysis level" from the gateway **200** drop-down menu, to select the level of analysis in a multi-level analysis sequence to be accomplished at the local level by a gateway-resident analysis engine. The user accesses the container editor to containerize (Fig. **5**). Input is received from the user selecting the registers by clicking on the graphically depicted register, or from a drop down menu.). Input is received from the user setting the registers as described elsewhere in ("container registers"). Input is received from the user selecting or defining the rules governing the interaction of that gateway with transient containers. Input is received from the user selecting or defining the rules governing the interaction of containers existing within the logical domain of the container 100 to which that gateway is attached. The user publishes the gateway (Fig. **5**). Input is received from the user selecting "residence" from the main menu bar.

). Input is received from the user choosing to leave the gateway where it was created, move it to container on another drive, directory, computer, or network. If the user chooses "move," a browse function allows the user to name the new location or browse a list of possible locations. Once input is received from the user choosing the residence of the gateway, the program restores the search interface screen.

The invention includes a data reporting means editor **610**, and a data collection means editor **710,** Fig. 2 A, as a menu option under the gateway editor **210**.

The present invention also includes a gateway process.

Referring now to FIG. **12**, a flow chart of the gateway process is shown. A system operation, search process or element container or process container is shown in transit **21100** passing through a gateway **21200**. The container, operation or process interacts with the gateway **21300**, uploading, downloading and exchanging information with the container, operation or process. The gateway stores container information **21400** and the container registers are altered **21500**. The container registers also interact with the search interface **21600**. The gateways report the register information or make it available for collection by the data reporting and collection means (Fig. **8**) operating on the communication device **26** to provide the information to the analysis engine **21800**, which stores **90100**, analyzes and instructs the execution engine **21900**, which processes and instructions are also stored **90100** by the execution engine upon receipt.

All operations in Fig. **12** utilize the communication device **26** whenever necessary or expeditious.

Referring now to FIG. **13 A,** a drawing of nested containers **100** prior to the container modification process on a network **201** is shown. (Note: The same container numbering scheme is used in Fig. 13 A, 13 B, 13 C, 13 D and in 2 B.) Information containers **505** and **909**, residing within container **908**, operating under the rules governing container interaction within that container **908** downloaded to container **505** and **909** from gateway **9081** upon their entrance to container **908,** which rules had been downloaded from execution engine **500** acting under the direction of analysis engine **400**, and under the rules programmed into their own registers **404120**, **909120**, compare the specified (by those rules) set of registers **404120**, **909120**, i.e., time and space, and determine a container **404** encapsulated within **505** would be more appropriately encapsulated within container **909**.

Referring now to FIG. **13 B** a drawing of nested containers during a container modification process on a network **201** is shown. Container **404** is moved to reside with container **909**. As the container **404** exits container **505**, the gateway of container **505**, being gateway **5051**, operating under the rules governing container interaction with a gateway 5051 upon egress or egress as programmed in the gateway editor **210** and modified by the execution engine **500** executing the instructions of the analysis engine **400**, or any greater logical analysis engine **408** providing execution instructions to an execution engine **508** operating in a larger encompassing container **108** entering through that container's gateway **208** or an independent gateway **707**, or sub-analysis engine operating at any gateway level, records the register information of container **404.** The gateway **5051** reports the transaction to the gateway **9081** of container **908**, being the next higher logical container. Gateway **9081** holds in gateway storage **205** the information until collected by one or more data collection processes **700**, or reported to one or more data reporting processes **600**, serving one or more analysis engines **400** residing independently on the system **10** or an analysis engine at higher logical container **303**. The analysis engine **400,** comparing reports of user hierarchically graded usage under the operations of the search engine **320** and the search interface **300**, on information container **808** after receiving reports from the data reporting means of container **404** being moved to container **909** determines, i.e., that the number of time and space relevant containers residing within container **909** is sufficient to warrant an action, and directs the execution engine **500** to copy container **909**, nested within container **908**, to a third information container **808**. As the copy instruction from execution engine **500** transits the gateway of container **908**, the gateway **9081** records the instruction. The copy instruction interacts with the registers **909120** of container **909** regarding the rules governing its copying to another location. Once approved by the governing rules of registers **909120** appended to container **909**, container **909** is duplicated. As the duplicate container **909** exits the container **908**, the gateway records the register information **909120** of container **909**, and the registers **909120** of container **909** are altered by special instructions from gateway **9081** under the rules residing in gateway **9081** regarding ingress and egress and the rules residing in the registers **909120** of container **909** regarding alteration by gateways upon ingress and egress. Passing through independent gateway **707**, the register information **909120** is recorded, and awaits data collection or reporting 700, **600.** As container **909** enters container **808**, the gateway records the register information **909120** of container **909**, the registers **909120** of **909** are altered by special instructions from gateway **8081**, operating under the rules as described in the paragraph above, and container **909** takes up residence within container **808.**

Referring now to FIG. **13 C,** a drawing of nested containers after the container modification process on a network **201** process is shown. Container **909**, now also logically residing within container **808,** commences to interact with other containers **606** in **808** under the rules governing container interaction within container **808** as received from gateway **8081** upon transiting that gateway, and under the rules of registers **606120, 909120 of** the interacting containers **606, 909**, operating under the rules as described in the paragraph above. Through data collection and reporting **700, 600**, analysis engine is appraised of container's **909** new duplicate residence. I.e., operating under the registers of space relevance, a body of law pertaining to Boston Municipal tax law may be housed in a container holding Massachusetts tax law, but it would be more appropriately located in a container holding Boston tax law, with only a pointer to that location residing in the Massachusetts tax law container. In this example, such an analysis could be accomplished by comparison of zip code information in the space registers, or logical rules-based analysis, with "state" being a larger set than "city". Or, i.e., operating under the registers of time relevance, the curve of time relevance for a concert might follow an ascending curve for the months prior, hit a brief plateau, and then reach a precipitous decline, at which time certain pertinent information only might be moved to an archival container of city events or rock concerts of that year. In this example, once the curve is mapped into a register, that map would cause an increasing frequency of pointers to that container in other containers or gateways, or inclusion of that container in other containers, as the analysis engine compares that curve with increasing user inquiry.

Referring now to Fig. **13 D**, a flowchart of the reconstruction process is shown.

Information containers **505** and **909**, residing within container **908**, operating under the rules governing container interaction within that container **908** downloaded **888103** to container **505** and **909** from gateway **9081** upon their entrance to container **908,** which rules had been downloaded **888102** from execution engine **500** acting under the direction **888101** of analysis engine **400**, and under the rules programmed into their own registers **404120, 909120**, compare **888104** the specified (by those rules) set of registers **404120, 909120**, i.e., time and space, and determine **888105** a container **404** encapsulated within **505** would be more appropriately encapsulated within container **909**.

Container **404** is moved **888106** to reside with container **909**. As the container **404** exits container **505**, the gateway of container **505**, being gateway **5051**, operating under the rules governing container interaction with a gateway **5051** upon egress or egress as programmed in the gateway editor **210** and modified **888108** by the execution engine **500** executing the instructions of the analysis engine **400**, or any greater logical analysis engine **408** providing execution instructions **888107** to an execution engine **508** operating in a larger encompassing container **108** entering through that container's gateway **208** or an independent gateway **707**, or sub-analysis engine operating at any gateway level, records **888109** the register information of container **404**, and alters the register information of container **404**. The gateway **5051** reports **888110** the transaction to the gateway **9081** of container **908**, being the next higher logical container. Gateway **9081** holds **888111** in gateway storage **205** the information until collected by one or more data collection processes **700**, or reported to one or more data reporting processes **600**, serving **888112** one or more analysis engines **400** residing independently on the system **10** or an analysis engine at higher logical container **303**. The analysis engine **400**, comparing **888114** reports of user hierarchically graded usage on information container **808** under the operations of the search engine **320** and the search interface **300**, after receiving **888113** reports from the data reporting means of container **404** being moved to container **909**, determines **888115**, i.e., that the number of time and space relevant containers residing within container **909** is sufficient to warrant an action, and directs **888115** the execution engine **500** to copy container **909**, nested within container **908**, to a third information container **808**. As the copy instruction from execution engine **500** transits the gateway of container **908**, the gateway **9081** records **888116** the instruction. The copy instruction interacts **888117** with the registers **909120** of container **909** regarding the rules governing its copying to another location. Once approved **888118** by the governing rules of registers **909120** appended to container **909**, container **909** is duplicated **888118**. As the duplicate container **909** exits the container **908**, the gateway records **888119** the register information **909120** of container **909**, and the registers **909120** of container **909** are altered **888120** by special instructions from gateway **9081** under the rules residing in gateway **9081** regarding ingress and egress and the rules residing in the registers **909120** of container **909** regarding alteration by gateways upon ingress and egress. Passing through independent gateway **707**, the register information **909120** is recorded **888121**, and awaits **888122** data collection or reporting **700, 600**. As container **909** enters container **808**, the gateway records **888123** the register information **909120** of container **909**, the registers **909120** of **909** are altered **888124** by special instructions from gateway **8081**, operating under the rules as described in the paragraph above, and container **909** takes up residence **888125** within container **808.**

Container **909**, now also logically residing (in addition to its original container residence) within container **808,** commences to interact **888126** with other containers **606** in **808** under the rules governing container interaction within container **808** as received from gateway **8081** upon transiting that gateway, and under the rules of registers **606120, 909120 of** the interacting containers **606, 909**, operating under the rules as described in the paragraph above. Through data collection and reporting **700**, **600**, analysis engine is appraised **888127** of container's **909** new duplicate residence.

Referring now to Fig. **14**, the screen interface of the container editor is shown. This interface is a process wherein input is received by the user using the main menu **78** or drop down menu **1419**, or using an input device to "drag and drop" or click, causing the system **10** to acquire **1409**, edit **1410** or create **1411** a file **1407**, container **1408** or digital content **01,** to search for **1412**, acquire **1413**, edit **1414** or create **1415**, print **1416**, or containerize **1417** a container **100**, to select **1402**, (or by clicking on register), search **1403**, acquire **1404**, edit **1405**, or create a register **1406** to append or detach registers **120** to those containers, to set register values in those registers **120**, to utilize the register editor **125** through **1405** to create new registers, or to **1418** add, detach, acquire a gateway **200** to append or detach to those containers, and utilize the gateway editor **210** through **1418**. (See detailed description referring to Fig. 5)

Referring now to Fig. **15**, the screen interface of the gateway editor is shown. This interface is a process wherein input is received by the user using the main menu **1501** or drop down menu **1513**, or using an input device to "drag and drop" or click, causing the system **10** to search for **1507**, acquire **1508**, edit **1509** create **1510**, print **1511** or containerize **1512** gateways, and causing the system **10** to establish rules by which an individual gateway governs the transiting **1502**, entering **1503**, exiting **1504** of containers and the interaction of containers within its domain **1505**, and external of its domain.**1506**. (See detailed description referring to Fig. **11**).

Referring now to Fig. **16**, the screen interface of the search interface. This interface is a process wherein input is received by the user using the main menu **1625** or drop down menu

1624, or using an input device to "drag and drop" or click, or by entering text, causing the system **10** to select **1615**, search for **1616**, acquire **1617**, edit **1618** create **1619**, print **1620**, containerize **1621** (by accessing the container editor **110)** or insert **1622** digital search strings into the search box **1623** in order to submit that string to the search engine **320**, or causing the system **10** to select **1602**, search for **1603**, acquire **1604**, edit **1605**, create **1612**, containerize **1613** (by accessing the container editor **110**), or insert **1614** search keys (templates that comprise search scope in geographic range, container level, and specific key words or digital strings), or containerized searches (containers **110**), into the search box **1623** in order to submit that string to the search engine **320** , or causing the system **10** to set a search range by geographic range **1607**, container level **1608**, or acquire **1609**, edit **1610** or create **1611** a scope template. (templates that comprise search scope in geographic range and, container level.) (See detailed description referring to Fig. 6).

Referring now to Fig. **17**, a drawing showing, on an input device or computer screen **24**, in any generic (dashed lines) software application program, a drop-down menu link **1403** on a drop down menu **1402** dropping down from a main menu **1401**, and a free-floating button link **1404**, is shown. When input is received at **1402** or **1403**, the system **10** makes available to the user the containerization process or container editor **110.** When input is received at drop-down menu link **1405** or a button link **1406**, the system **10** makes available to the user the means to enter and interact with this system **10** or this network **201** in any of their aspects. The interfaces **1403, 1404** show a process wherein input is received causing the system **10** to encapsulate content or access the container editor **110**. The link also allows the user to encapsulate the page or file on which he is currently working, without selecting content, and if so desired, without accessing the container editor. The interfaces **1405, 1406** show a process wherein input is received causing the system **10** to access or interact with the system **10** or the network **201**.

The present invention also includes a search engine **320**. Once the key word(s), phrase or digital segment is entered into the search interface **300,** or an offered selection chosen on the menu, it is utilized by the search engine **320** to locate the desired site or data.

The search engine employed may be any industry standard search engine such as Verity "Topic", or Personal Library Software, as used in Dow Jones News Retrieval, or Internet search engines such as Webcrawler, Yahoo, Excite, Infoseek, Alexa or any Internet search engine, or any new engines to be developed capable of searching for and locating digital segments, whether text, audio, video or graphic.

The present invention also includes an analysis engine **400.** Utilizing rules-based analysis, the analysis engine determines the class of storage medium upon which containers reside, the subsets and supersets by which and in which containers encompass and reside within one another, the routes of access to those containers, the historically successful search parameters by which those containers are accessed based upon the identity of the user accessing the containers, and the grade of access chosen by the user in accessing that container **100**.

Utilizing a pre-programmed sequence of compilation, and inductive, deductive and derivative analysis, the analysis engine manufactures instructions based upon the analysis of the information submitted by the gateways and the search interface, and submits those instructions to the appropriate execution engine **500** in order to create new information containers, content assemblages, storage schemes, access routes, search templates, and gateway instructions, and others, and to provide informed search options through the search interface to the inquiring user.

The present invention also includes an engine editor **510**, that provides a system administrator with a means of editing the operating principles of that search engine, and search template loading in the search interface **300**, a reporting and collection means editor **610**, **710**, governing data reporting **600** and data collection **700** at the gateways **200** as defined by the gateway editor **210** and the register editor **125**, a container editor **110** for creating and modifying containers and appending registers to containers, a register editor **125** for creating and modifying container registers and establishing and adjusting the values therein, container gateways **200** with their own storage **205**, information containers **100** for holding information and container registers for holding information about specific containers and their history on the network.

The present invention also includes an execution engine **300**. Based upon instructions received from the analysis engine **400** utilizing the communication device **26**, the execution engine **500** provides search phrases to the search interface **300** based upon initially received inquiries, relocates containers including their programs, data and registers to other directories, drives, computers, networks on other classes of storage mediums, i.e., tape drive, optical drive, CD-ROM, deletes, copies, moves containers to nest within or encompass other containers on other directories, drives, computers, networks to nest within other containers, alters the class of storage medium upon which containers reside, the subsets and supersets by which and in which containers encompass and reside within one another, the routes of access to those containers, and the historically successful search parameters by which those containers are accessed based upon the identity of the user accessing the container and the grade of access chosen by the user in accessing that container.

The execution engine **400** fulfills the instructions of the analysis search engine **500**, to create new information containers, content sub and superset assemblages, storage schemes, access routes, search templates, gateway **200** instructions and other system functions. The execution engine includes an editor **510** that provides a system manager with a means of editing the operating principles of that search engine, governing data reporting, data collection **700**, search template loading, gateway instructions, and other functions.

The present invention also includes flat or relational databases **900**, used where, and as required.

The present invention also includes a communication device **26** supporting all operations on a network wide basis.

The present invention also includes a search engine **300** to locate the desired site or data. The present invention also includes databases **900**, flat or relational, to serve the other components of the system as needed and where needed.

The present invention also includes editors, by which the user may alter the governing aspects of the system. Editors include, but are not limited to, a container editor **110**, a register editor **125**, a gateway editor.**210**, an engine editor **510**, a reporting means editor **610**, a search interface **300**, and a collection means editor **710**.

The present invention also includes specific screen interfaces for the editors, as described in Fig. **14**, Fig. **15**. and Fig. **16**.

The present invention also includes a means for this system **10** and network **201 or container editor 110** to be accessed from a menu or button selection within any program, as described in Fig. **17**.

While the present invention has been described with reference to certain preferred embodiments, those skilled in the art will recognize that various modifications may be provided. For example, both analysis engine and execution engine may be duplicated or modified for distribution at various locations and hierarchical positions in the gateway and container system throughout the network and designed to work in concert. Also, the physical computing infrastructure may be mainframe, mini, client server or other with various network and distributed computing designs, including digitally supported or based physical or public media, and the components of the system 10, as described in Fig. 1 may be physically distributed through space. Even the contents of a single container may be logically referenced but be physically distributed through the network and reside at multiple storage locations. The whole system may be hierarchically nested within other systems to the nth degree. Whole systems may also be encapsulated within containers. A single container may also encompass a single physical media, such as a CD-ROM disk, programmed with the container, gateway and register design. Gateways may be strategically placed on containers at ingress and/or egress points or may be placed strategically throughout the system for optimal collection and reporting output and gateway system control. Also, the loop of gateway data collection and reporting, analysis engine analysis, instruction, and gateway modification, and execution engine operations may be infinitely nested, from the smallest container of two sub-containers to whole networks holding millions of containers and thousands of levels, with analysis itself nested within the multiple levels. Gateways may be established at both logical and physical junctures such as a satellite uplink point. Also, the provision to establish a unique network identity might be designed to include as of yet unknown computer networks as they arise. The analysis and execution engines may operate on a rules-based, fuzzy logic, artificial intelligence, neural net, or other system not yet devised. Other variations upon and modifications to the preferred embodiments are provided for by the present invention, which is limited only by the following claims. Also, the classification scheme of nested containers, while designated by the container creators, may transform, be utilized otherwise, or be wholly discarded according to usage. Also, hardware configurations, such as the use of RAM or hard drives for storage or lasers for communication may assume myriad forms without altering the essential operation of this invention.

WHAT IS CLAIMED IS:

Sub a1

1. An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element;

a plurality of registers, the plurality of register forming part of the container, a first register of the plurality of registers for storing a unique container identification value, a second register of the plurality of registers that stores information and evolves according to the relationship, use and interaction of the container with other containers, processes and systems; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems and process.

2. The apparatus of claim 1, wherein the information element is one from the group of text, graphic images, video, audio, a digital pattern, a process, a nested container, bit, natural number and a system.

3. The system of claim 1, wherein the plurality of registers include at least one container history register for storing information regarding past interaction of the container with other container, system or processes, the container history register being modified.

4. The system of claim 1, wherein the plurality of registers include at least one system history register for storing information regarding past interaction of the container with different operating system and network processes.

5. The system of claim 1, wherein the plurality of registers include at least one predefined register, the predefined register being a register associated with an editor for user selection, the predefined register appendable to any container.

6. The system of claim 1, wherein the plurality of registers include a user-created register, the user-created register generated by the user, one or more of which is appendable to any container.

7. The system of claim 1, wherein the plurality of registers include a system-defined register, the system-defined register set, controlled and used by the system, one or more of which is appendable to any container.

8. The system of claim 1, wherein the plurality of registers include at least one register for controlling the relationship of the container with other containers, systems and processes using time as a parameter.

9. The system of claim 8, wherein the plurality of registers include:

an active time register for identifying times at which the container will act upon other containers, processes, systems or gateways;

an passive time register for identifying times at which the container can be acted upon other containers, processes systems, or gateways; and

a neutral time register for identifying times at which the container may interact with other containers.

10. The system of claim 1, wherein the plurality of registers include at least one acquire register for controlling whether the container adds a register or a container from other containers when interacting with them.

11. The system of claim 1, wherein the plurality of registers include at least one register for controlling the relationship of the container with other containers using space as a parameter.

12. The system of claim 11, wherein space refers to the geographic location of a the container.

13. The system of claim 11, wherein space refers to the logical address space of a network in which a container resides.

14. The system of claim 11, wherein the plurality of registers include:

an active space register for identifying space in which the container will act upon other containers, processes, systems or gateways;

an passive space register for identifying from which the container can be acted upon other containers, processes systems, or gateways; and

a neutral time register for identifying space in which the container may interact with other containers.

15. The system of claim 1, wherein the gateway includes means for acting upon another container, the means for acting upon another container using the plurality of register to determine whether and how the container acts upon other containers.

16. The system of claim 1, wherein the gateway includes means for allowing interaction, the means for allowing interaction using the plurality of registers to determine whether and how another container can act upon the container.

17. The system of claim 1, wherein the gateway includes means for gathering information, the means for gathering information recording register information from other containers, systems and processes that interact with the container.

18. The system of claim 1, wherein the gateway includes means for reporting information, the means for reporting information providing register information to other containers, systems and processes that interact with the container.

19. The system of claim 1, wherein the gateway includes an expert system including rules defining the interaction of the container with other containers, systems and processes.

20. A method for creating an interactive information container, the method including the steps of:

forming a container;

selecting an interactive register for the container;

identifying an item for inclusion in a container; and

creating a container element that includes the identified item.

21. The method of claim 20, wherein the step of forming a container further comprising the steps of:

displaying a plurality of container levels;

receiving input from a user selecting one of the displayed container level; and

displaying a container template corresponding to the container level input.

22. The method of claim 20, wherein the step of selecting an interactive register further comprising the steps of:

displaying a list of available registers;

receiving input selecting an available register from the list of available registers;

receiving input values for the selected available register; and

appending the register to the container.

23. The method of claim 20, wherein the step of creating a container element that includes the identified item further comprising the steps of:

providing a data structure as part of the container element;

storing the identified element in the data structure; and

associating the container element with the container.

24. The method of claim 20, wherein the step of forming a container includes the step of providing for the container a gateway that uses the interactive register to control the interaction of the container with other containers, processes, and systems.

25. The method of claim 24, wherein the step of providing a gateway further comprising the steps of:

determining a current gateway for a system upon which the container is being created;

replicating the current gateway to create a new gateway; and

associating the new gateway with the container.

26. The method of claim 24, wherein the step of providing a gateway further comprising the steps of:

determining a register for a system upon which the container is being created;

replicating the determined register to create a new register; and

associating the new register with the container.

27. The method of claim 20, wherein the step of selecting an interactive register further comprising the steps of:

retrieving a list of available registers;

selecting an available register from the list of available registers by the system;

receiving input values for the selected available register from the system; and

appending the register to the container.

28. The method of claim 20, wherein the step of creating a container element that includes the identified item is performed by a system interacting with the container, and further comprising the steps of:

providing a data structure as part of the container element;

storing the identified element in the data structure; and

associating the container element with the container.

29. A method for interacting between a first interactive information container and a second interactive information container, the method including the steps of:

determining identification information for the first container using a first gateway;

determining identification information for the second container using a second gateway;

determining whether the first container can act upon the second container using the first gateway and a register of the first container;

determining whether the second container can be acted upon by the first container using the second gateway and a register of the second container; and

performing the interaction between the first and second containers prescribed by the first gateway and the register of the first container if both the first container can act upon the second container and the second container can be acted upon by the first container.

30. The method for interacting of claim 29, wherein the steps of determining identification information are performed by reading respective identification registers of the first and second containers.

31. The method for interacting of claim 29, further comprising the step of altering a register of the first container ~~and~~ a register of the second container to reflect the interaction between the first container and the second container.

32. The method for interacting of claim 29, further comprising the step of adding registers to the first container based on the registers in the second container and the second gateway.

33. The method for interacting of claim 29, wherein the step of performing also uses the second gateway and the register of the second container to determine the prescribe action to be taken.

34. The method for interacting of claim 29, further comprising the steps of:

determining whether the first container should add an identified register of the second container as a new register of the first container using an acquire register and the first gateway of the first container; and

adding the new register to the first container if it is determined that the new register should be added to the first container.

35. The method for interacting of claim 29, further comprising the step of modifying the first gateway of the first container based on the interaction between the first container and the second container.

36. The method for interacting of claim 35, wherein the step of modifying includes modifying rules of an expert system that forms the first gateway of the first container.

Sub a'7

PTO/SB/01 (6–95) (modified)
Approved for use through 10/31/96 OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

0010/PTO
Rev. 6/95

U.S. Department of Commerce
Patent and Trademark Office

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION

[X] Declaration Submitted with Initial Filing OR [ ] Declaration Submitted after Initial Filing

Attorney Docket Number	**3726**
First Named Inventor	**Michael De Angelo**
COMPLETE IF KNOWN	
Application Number	**Unknown**
Filing Date	**Even Date Herewith**
Group Art Unit	**Unknown**
Examiner Name	**Unknown**

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

the specification of which (*Title of the Invention*)

[ ] is attached hereto

OR

[X] was filed on (MM/DD/YYYY) **[01/28/1999]** as United States Application Number or PCT International Application Number **[PCT/US99/01988]** and was amended on (MM/DD/YYYY) [________________] (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37 Code of Federal Regulations. § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code § 119 (a)-(d) or § 385(b) of any foreign application(s) for patent or inventor's certificate, or § 365 (a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]

[ ] Additional foreign application numbers are listed on a supplemental priority sheet attached hereto:

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	
60/073,209	**01/30/1998**	[ ] Additional provisional application numbers are listed on a supplemental sheet attached hereto.

DECLARATION	**Page 2**

I hereby claim the benefit under Title 35, United States Code § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (*if applicable*)
	PCT/US99/01988	**01/28/1999**	

[ ] Additional U.S. or PCT international application numbers are listed on a supplemental priority sheet attached hereto.

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Name	Registration Number	Name	Registration Number
Greg T. Sueoka **James K. Okamoto**	33,800 40,110		

[ ] Additional attorney(s) and/or agent(s) named on a supplemental sheet attached hereto.

2

Please direct all correspondence to:

Greg T. Sueoka
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
U.S.A.

Telephone	**(650) 858-7194**	Fax	**(650) 494-1417**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

1-00

Name of Sole or First Inventor:	[ ] A petition has been filed for this unsigned inventor							
Given Name	**Michael**	Middle Initial		Family Name	**De Angelo**	Suffix e.g. Jr.		
Inventor's Signature	Michael DeAngelo			Date	April 5, 1999			
Residence: City	**Santa Barbara**	State	**CA** CA	Country	**USA**	Citizenship		
Mailing Address	**1324 J State Street, Suite 290**							
Mailing Address								
City	**Santa Barbara**	State	**CA**	Zip	**93101**	Country	**USA**	

[ ] Additional inventors are being named on supplemental sheet(s) attached hereto

Rev. 04/01/99

PTO/SB/ 10 (6-95) (modified)
Approved for use through 07/31/96. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(c))--SMALL BUSINESS CONCERN

Docket Number (Optional): 3726

Applicant or Patentee: Michael De Angelo
Application or Patent No.:
Filing Date or Issue Date:
Title: System And Method For Creating And Manipulating Information Containers With Dynamic Registers

I hereby declare that I am

[ ] the owner of the small business concern identified below:
[X] an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN Ematrix Corporation
ADDRESS OF SMALL BUSINESS CONCERN 104 West Anapamu, Suite C
Santa Barbara, California 93101

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

[X] the specification filed herewith with title as listed above.
[ ] the application identified above.
[ ] the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each such person, concern or organization having any rights in the invention is listed below:

[X] No such person, concern, or organization exists.
[ ] Each such person, concern or organization is listed below:

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING Michael De Angelo
TITLE OF PERSON IF OTHER THAN OWNER Officer
ADDRESS OF PERSON SIGNING 104 West Anapamu, Suite C, Santa Barbara, California 93101
SIGNATURE Michael DeAngelo DATE April 5, 1999

Rev. 04/01/99

Bib Data Sheet

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

SERIAL NUMBER 09/284,113	**FILING DATE** 04/07/1999 **RULE** _	**CLASS** 707	**GROUP ART UNIT** 2771	**ATTORNEY DOCKET NO.** 3726-US

APPLICANTS

MICHAEL DE ANGELO, SANTA BARBARA, CA UNITED STATES;

**** CONTINUING DATA ***************************** Yes LW

THIS APPLICATION IS A 371 OF PCT/US99/01988 01/28/1999
WHICH CLAIMS BENEFIT OF 60/073,209 01/30/1998

**** FOREIGN APPLICATIONS *********************** None

IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 04/12/2000 **

Foreign Priority claimed □ yes ☑ no 35 USC 119 (a-d) conditions met □ yes ☑ no □ Met after Allowance Verified and Acknowledged ____ Examiner's Signature LW Initials	**STATE OR COUNTRY** CA	**SHEETS DRAWING** 30	**TOTAL CLAIMS** 36	**INDEPENDENT CLAIMS** 3

ADDRESS

GREG T SUEOKA
FENWICK & WEST
TWO PALO ALTO SQUARE
PALO ALTO ,CA 94306

TITLE

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

FILING FEE RECEIVED 524	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other ____________ □ Credit

```
                    DO/EO BIBLIOGRAPHIC DATA ENTRY

SERIAL NUMBER:      09 /  284113                RECEIPT DATE:      04 /  07 /  99
IA NUMBER:  PCT/  US99 /  01988                 IA FILING DATE:    01 /  28 /  99
FAMILY NAME:        DE ANGELO                   DELAY WAIVED (Y/N):              Y
GIVEN NAME:         MICHAEL                     DEMAND RECEIVED (Y/N):       ~~Y~~ N
PRIORITY CLAIMED (Y/N):     Y                   PRIORITY DATE:     01 /  30 /  98
NO BASIC FEE (Y/N):         N                   US DESIGNATED ONLY (Y/N):        N
ATTORNEY DOCKET NUMBER:     3726 US             COUNTRY:          USX
CORRESPONDENCE NAME/ADDRESS:  CUSTOMER NUMBER:  000000  TELEPHONE  0000000000
                                                        FAX        0000000000
NAME:     GREG T SUEOKA
          FENWICK & WEST
STREET:   TWO PALO ALTO SQUARE

CITY:     PALO ALTO
STATE/COUNTRY:  CA      ZIP:  94306
EMAIL:
APPLICATION TITLES:
   SYSTEM AND METHOD FOR CREATING AND MANIPULATING
   INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

                    TAB TO LAST POSITION,PUSH SEND
```

PATENT APPLICATION SERIAL NO. 09/284113

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

04/12/1999 WCLAYBRO 00000037 09284113

01 FC:967	144.00 OP
03 FC 959	380.00 OP

PTO-1556
(5/87)

PATENT APPLICATION FEE DETERMINATION RECORD
Effective November 10, 1998

Application or Docket Number
09/284113

CLAIMS AS FILED - PART I

SMALL ENTITY TYPE ☑ OR OTHER THAN SMALL ENTITY

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE	FEE		RATE	FEE
BASIC FEE			380	380.00	OR		~~700.00~~
TOTAL CLAIMS	36 minus 20=	* 16	X$ 9=	$144	OR	X$18=	
INDEPENDENT CLAIMS	3 minus 3 =	*	X39=		OR	X78=	
MULTIPLE DEPENDENT CLAIM PRESENT			+130=		OR	+260=	
			TOTAL	$524	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT A		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
	Total	* 16	Minus	** 36	= —	X$ 9=		OR	X$18=	
	Independent	* 2	Minus	*** 3	= —	X39=		OR	X78=	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
						TOTAL ADDIT. FEE	∅	OR	TOTAL ADDIT. FEE	

AMENDMENT B		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
	Total	*	Minus	**	=	X$ 9=		OR	X$18=	
	Independent	*	Minus	***	=	X39=		OR	X78=	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
						TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
	Total	*	Minus	**	=	X$ 9=		OR	X$18=	
	Independent	*	Minus	***	=	X39=		OR	X78=	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
						TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

☆U.S. GPO:1998-454-473/90301

MULTIPLE DEPENDENT CLAIM FEE CALCULATION SHEET
(FOR USE WITH FORM PTO-875)

SERIAL NO. 09/284113

FILING DATE

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT			*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.		IND.	DEP.	IND.	DEP.	IND.	DEP.
1	/						51						
2		/					52						
3		/					53						
4		/					54						
5		/					55						
6		/					56						
7		/					57						
8		/					58						
9		/					59						
10		/					60						
11		/					61						
12		/					62						
13		/					63						
14		/					64						
15		/					65						
16		/					66						
17		/					67						
18		/					68						
19		/					69						
20	/						70						
21		/					71						
22		/					72						
23		/					73						
24		/					74						
25		/					75						
26		/					76						
27		/					77						
28		/					78						
29	/						79						
30		/					80						
31		/					81						
32		/					82						
33		/					83						
34		/					84						
35		/					85						
36		/					86						
37							87						
38							88						
39							89						
40							90						
41							91						
42							92						
43							93						
44							94						
45							95						
46							96						
47							97						
48							98						
49							99						
50							100						
TOTAL IND.	3	↲		↲		↲	TOTAL IND.		↲		↲		↲
TOTAL DEP.	33						TOTAL DEP.						
TOTAL CLAIMS	36	▨		▨		▨	TOTAL CLAIMS		▨		▨		▨

PTO-1360 (3-78)

**MAY BE USED FOR ADDITIONAL CLAIMS OR AMENDMENTS*

U.S. DEPARTMENT of COMMERCE
Patent and Trademark Office

O/ EO \ ORKSHEET

U.S. Appl. No. 09/284113

International Appl. No. US99/01988

Application filed by : ☑ 20 months ☐ 30 months

WIPO PUBLICATION INFORMATION :

Publication No.: WO99/39285

Publication Date : 8/5/99

Publication Language : English

Not Published : ☐ U.S. only designated
☐ EP request

Screening Done by :
Barbara Campbell
National Stage Processing
(703) 305-3631

INTERNATIONAL APPLICATION PAPERS IN THE APPLICATION FILE :

☑ International Application *(RECORD COPY)*
☐ Article 19 Amendments
☐ PCT/IB/331
☐ PCT/IPEA/409 IPER (PCT/IPEA/416 on front)
☐ Annexes to 409
☐ Priority Document (s) No. _____

☐ International Appl. on Double Sided Paper *(COPIES MADE)*
☐ Request form PCT/RO/101
☑ PCT/ISA/210 - Search Report
☐ Search Report References
☐ Other : ____________________

RECEIPTS FROM THE APPLICANT *(other than checked above)* :

☑ Baisc National Fee *(paid or authorized to charge)*
☑ Description
☑ Claims
☑ Words in the Drawing Figure(s)
☐ Article 19 Amendments
☐ Annexes to 409
☐ entered ☐ not entered
☑ Oath/ Declaration (executed)
☐ DNA Diskette

☐ Preliminary Amendment(s) Filed on : ________ ________
☐ Information Disclosure Statement(s) Filed on : ________ ________
☑ Assignment Document
☑ Power of Attorney/ Change of Address
☐ Substitute Specification Filed on : ________ ________
☑ Verified Small Status Claim
(If submitted after Receipt Date - Is it timely? [✓ Y]/N)
☐ Other : ____________________

NOTES : Used IA

U.S.C. 371 - Receipt of Request (PTO-1390)	April 7, 1999
te Acceptable Oath/ Declaration Received	"
:e Complete 35 U.S.C. 371	"
:(e) Date	"
e of Completion of DO/ EO 906 - Notification of Missing 102(e) Requirements	
e of Completion of DO/ EO 907 - Notification of Acceptance for 102(e) Date	
e of Completion of DO/ EO 911 - Application Accepted Under 35 U.S.C. 111	
e of Completion of DO/ EO 905 - Notification of Missing Requirements	
e of Completion of DO/ EO 916 - Notification of Defective Response	
: of Completion of DO/ EO 903 - Notification of Acceptance	Jan. 12, 2000
: of Completion of DO/ EO 909 - Notification of Abandonment	

PATENT COOPERATION TREATY

From the INTERNATIONAL BUREAU

To:

Assistant Commissioner for Patents
United States Patent and Trademark
Office
Box PCT
Washington, D.C.20231
ÉTATS-UNIS D'AMÉRIQUE

in its capacity as elected Office

PCT

NOTIFICATION OF ELECTION

(PCT Rule 61.2)

Date of mailing (day/month/year) 13 December 1999 (13.12.99)	
International application No. PCT/US99/01988	**Applicant's or agent's file reference** 3726PCT
International filing date (day/month/year) 28 January 1999 (28.01.99)	**Priority date** (day/month/year) 30 January 1998 (30.01.98)
Applicant DE ANGELO, Michael	

1. The designated Office is hereby notified of its election made:

☒ in the demand filed with the International Preliminary Examining Authority on:

23 July 1999 (23.07.99)

□ in a notice effecting later election filed with the International Bureau on:

2. The election ☒ was

□ was not

made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO **34, chemin des Colombettes** **1211 Geneva 20, Switzerland** Facsimile No.: (41-22) 740.14.35	Authorized officer Antonia Muller Telephone No.: (41-22) 338.83.38

09/284113 EV

PCT/US99/01988

PATENT COOPERATION TREATY

5610 5632 2700 2771

From the INTERNATIONAL BUREAU

PCT

To:

SUEOKA, Greg, T.
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
ÉTATS-UNIS D'AMÉRIQUE

INFORMATION CONCERNING ELECTED OFFICES NOTIFIED OF THEIR ELECTION

(PCT Rule 61.3)

Date of mailing (day/month/year) 13 December 1999 (13.12.99)		
Applicant's or agent's file reference 3726PCT	**IMPORTANT INFORMATION**	
International application No. PCT/US99/01988	**International filing date** (day/month/year) 28 January 1999 (28.01.99)	**Priority date** (day/month/year) 30 January 1998 (30.01.98)
Applicant EMATRIX CORPORATION et al		

1. The applicant is hereby informed that the International Bureau has, according to Article 31(7), notified each of the following Offices of its election:

   AP :GH,GM,KE,LS,MW,SD,SZ,UG,ZW
   EP :AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE
   National :AU,BG,BR,CA,CN,CZ,DE,IL,JP,KP,KR,MN,NO,NZ,PL,RO,RU,SE,SK,US

2. The following Offices have waived the requirement for the notification of their election; the notification will be sent to them by the International Bureau only upon their request:

   EA :AM,AZ,BY,KG,KZ,MD,RU,TJ,TM
   OA :BF,BJ,CF,CG,CI,CM,GA,GN,GW,ML,MR,NE,SN,TD,TG
   National :AL,AM,AT,AZ,BA,BB,BY,CH,CU,DK,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IN,IS,KE,KG,KZ,LC,LK,LR,LS,LT,LU,LV,MD,MG,MK,MW,MX,PT,SD,SG,SI,SL,TJ,TM,TR,TT,UA,UG,UZ,VN,YU,ZW

3. The applicant is reminded that he must enter the "national phase" **before the expiration of 30 months from the priority date** before each of the Offices listed above. This must be done by paying the national fee(s) and furnishing , if prescribed, a translation of the international application (Article 39(1)(a)), as well as, where applicable, by furnishing a translation of any annexes of the international preliminary examination report (Article 36(3)(b) and Rule 74.1).

   Some offices have fixed time limits expiring later than the above-mentioned time limit. For detailed information about the applicable time limits and the acts to be performed upon entry into the national phase before a particular Office, see Volume II of the PCT Applicant's Guide.

   The entry into the European regional phase is postponed **until 31 months from the priority date** for all States designated for the purposes of obtaining a European patent.

The International Bureau of WIPO **34, chemin des Colombettes** **1211 Geneva 20, Switzerland** Facsimile No. (41-22) 740.14.35	Authorized officer: Antonia Muller Telephone No. (41-22) 338.83.38

M.H. 5040 09/284113

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference 3726 PCT	**FOR FURTHER ACTION** See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)	
International application No. PCT/US99/01988	International filing date *(day/month/year)* 28 JANUARY 1999	Priority date *(day/month/year)* 30 JANUARY 1998
International Patent Classification (IPC) or national classification and IPC Please See Supplemental Sheet.		
Applicant EMATRIX CORPORATION		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 4 sheets.

   □ This report is also accompanied by ANNEXES, i.e., sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority. (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

   These annexes consist of a total of 0 sheets.

3. This report contains indications relating to the following items:

   I ☒ Basis of the report

   II □ Priority

   III □ Non-establishment of report with regard to novelty, inventive step or industrial applicability

   IV □ Lack of unity of invention

   V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

   VI □ Certain documents cited

   VII □ Certain defects in the international application

   VIII □ Certain observations on the international application

Date of submission of the demand 23 JULY 1999	Date of completion of this report 26 NOVEMBER 1999
Name and mailing address of the IPEA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer RUAY LIAN HO
Facsimile No. (703) 305-3230	Telephone No. (703) 305-3834

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US99/01988

I. Basis of the report

1. This report has been drawn on the basis of *(Substitute sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments):*

☒ the international application as originally filed.

☒ the description, pages (See Attached) , as originally filed.
pages ____________ , filed with the demand.
pages ____________ , filed with the letter of ____________.
pages ____________ , filed with the letter of ____________.

☒ the claims, Nos. (See Attached) , as originally filed.
Nos. ____________ , as amended under Article 19.
Nos. ____________ , filed with the demand.
Nos. ____________ , filed with the letter of ____________.
Nos. ____________ , filed with the letter of ____________.

☒ the drawings, sheets/~~fig~~ (See Attached) , as originally filed.
sheets/~~fig~~ ____________ , filed with the demand.
sheets/~~fig~~ ____________ , filed with the letter of ____________.
sheets/~~fig~~ ____________ , filed with the letter of ____________.

2. The amendments have resulted in the cancellation of:

☒ the description, pages NONE.
☒ the claims, Nos. NONE.
☒ the drawings, sheets/~~fig~~ NONE.

□ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed, as indicated in the ~~Supplemental Box~~ Additional observations below (Rule 70.2(c)).

Additional observations, if necessary:
ONE

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US99/01988

V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. STATEMENT

Novelty (N)	Claims	1-36	YES
	Claims	NONE	NO
Inventive Step (IS)	Claims	1-36	YES
	Claims	NONE	NO
Industrial Applicability (IA)	Claims	1-36	YES
	Claims	NONE	NO

2. CITATIONS AND EXPLANATIONS

Claims 1-36 meet the criteria set out in PCT Article 33(2)-(4), because the prior art does not teach or fairly suggest by any prior art.

The claimed features, such as a plurality of registers with containers and a gateway attached to and forming part of the container, are not disclosed, taught or suggested by any prior art.

------------------- NEW CITATIONS -------------------
NONE

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No.
PCT/US99/01988

Supplemental Box
(To be used when the space in any of the preceding boxes is not sufficient)

Continuation of: Boxes I - VIII

Sheet 10

CLASSIFICATION:

The International Patent Classification (IPC) and/or the National classification are as listed below:
IPC(6): G06F 17/30, 3/14 and US Cl.: 707/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 101, 102, 103, 104, 200, 201, 202, 203, 204, 205, 206; 709/202, 203, 218, 228; 713/200, 201

I. BASIS OF REPORT:

This report has been drawn on the basis of the description,
pages, 1-42, as originally filed.
pages, NONE, filed with the demand.
and additional amendments:
NONE

This report has been drawn on the basis of the claims,
numbers, 1-36, as originally filed.
numbers, NONE, as amended under Article 19.
numbers, NONE, filed with the demand.
and additional amendments:
NONE

This report has been drawn on the basis of the drawings,
sheets, 1-30, as originally filed.
sheets, NONE, filed with the demand.
and additional amendments:
NONE

09/284173

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **G06F 17/30, 3/14**	**A1**	**(11) International Publication Number:** **WO 99/39285** **(43) International Publication Date:** 5 August 1999 (05.08.99)

(21) International Application Number: PCT/US99/01988

(22) International Filing Date: 28 January 1999 (28.01.99)

(30) Priority Data:
60/073,209 30 January 1998 (30.01.98) US

(71) Applicant *(for all designated States except US)*: EMATRIX CORPORATION [US/US]; 104 West Anapamu, Santa Barbara, CA 93101 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: DE ANGELO, Michael [US/US]; Suite 290, 1324 J State Street, Santa Barbara, CA 93101 (US).

(74) Agents: SUEOKA, Greg, T. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).

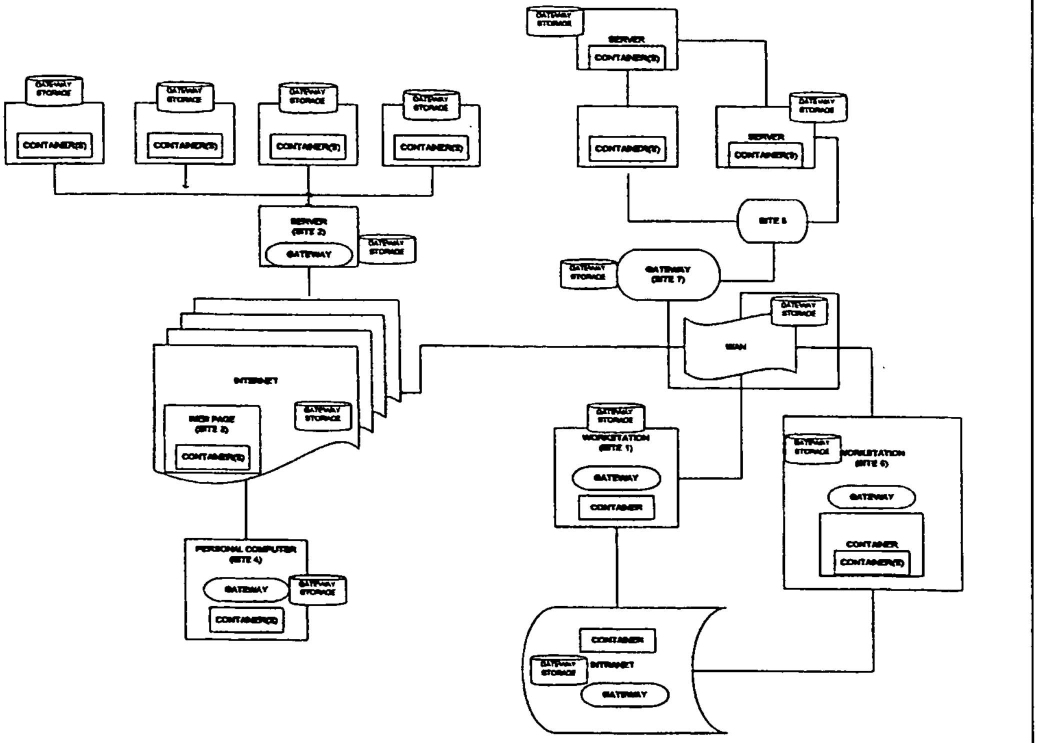
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

(57) Abstract

A system for creating and manipulating information containers with dynamic registers on a multi-user computer system, or computer network comprises an interactive information container, a container editor, a search interface, a user profile, system-wide hierarchical container gateways (site 7), interactive and evolving container registers, a data collection means, a data reporting means, an analysis engine with editor, an executing engine with editor, and a means of communicating with other computers, computer networks, or digital-based public or published media. The container editor provides an authoring user with the capacity to encapsulate any information component such as a file, set, database, network, event or process, and a set of parameters of multiple container registers to govern the interaction of that container with other containers or processes. The container registers include system-defined, system-alterable, user-defined and user-alterable registers.



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/01988

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 17/30, 3/14
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : Please See Extra Sheet.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

MICROSOFT COMPUTER DICTIONARY

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, PRO-QUEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,768,510 A (GISH et al) 16 June 1998, column 5.	1-36
A	US 5,848,246 A (GISH et al) 08 December 1998, column 5.	1-36

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

*	Special categories of cited documents:
"A"	document defining the general state of the art which is not considered to be of particular relevance
"E"	earlier document published on or after the international filing date
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O"	document referring to an oral disclosure, use, exhibition or other means
"P"	document published prior to the international filing date but later than the priority date claimed
"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&"	document member of the same patent family

Date of the actual completion of the international search 03 JUNE 1999	Date of mailing of the international search report 15 JUN 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer FoR RUAY LIAN HO Telephone No. (703) 305-3834

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/01988

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

707/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 101, 102, 103, 104, 200, 201, 202, 203, 204, 205, 206; 709/202, 203, 218, 228; 713/200, 201

B. FIELDS SEARCHED
Minimum documentation searched
Classification System: U.S.

707/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 101, 102, 103, 104, 200, 201, 202, 203, 204, 205, 206; 709/202, 203, 218, 228; 713/200, 201

CW

CORRECTED VERSION*

~~3640~~ 09/284113 2771

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30, 3/14

A1

(11) International Publication Number: WO 99/39285

(43) International Publication Date: 5 August 1999 (05.08.99)

(21) International Application Number: PCT/US99/01988

(22) International Filing Date: 28 January 1999 (28.01.99)

(30) Priority Data:
60/073,209 30 January 1998 (30.01.98) US

(71) Applicant (*for all designated States except US*): EMATRIX CORPORATION [US/US]; 104 West Anapamu, Santa Barbara, CA 93101 (US).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): DE ANGELO, Michael [US/US]; Suite 290, 1324 J State Street, Santa Barbara, CA 93101 (US).

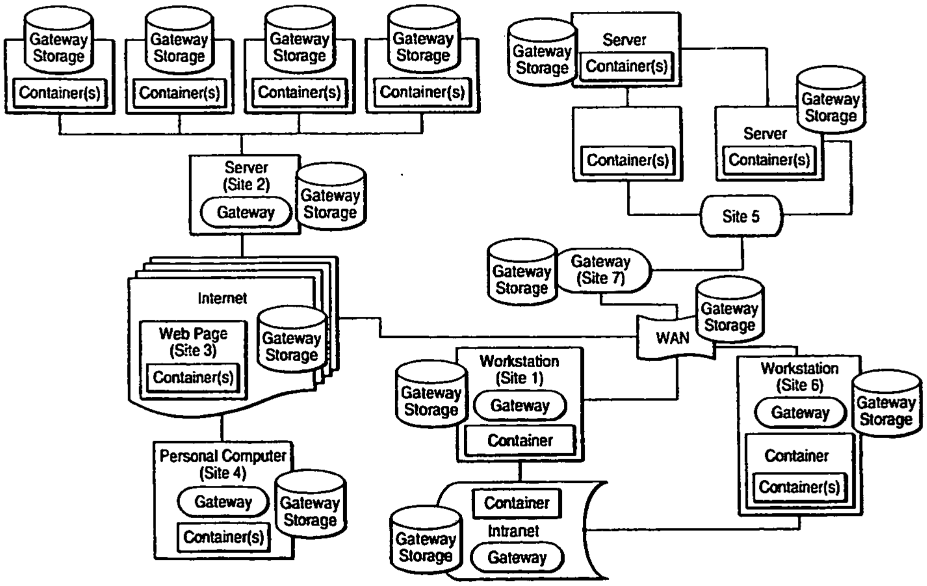
(74) Agents: SUEOKA, Greg, T. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

(57) Abstract



A system for creating and manipulating information containers with dynamic registers on a multi-user computer system, or computer network comprises an interactive information container, a container editor, a search interface, a user profile, system-wide hierarchical container gateways (site 7), interactive and evolving container registers, a data collection means, a data reporting means, an analysis engine with editor, an executing engine with editor, and a means of communicating with other computers, computer networks, or digital-based public or published media. The container editor provides an authoring user with the capacity to encapsulate any information component such as a file, set, database, network, event or process, and a set of parameters of multiple container registers to govern the interaction of that container with other containers or processes. The container registers include system-defined, system-alterable, user-defined and user-alterable registers.

*(Referred to in PCT Gazette No. 51/1999, Section II)

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to computer systems in a multi-user mainframe or mini computer system, a client server network, or in local, wide area or public networks, and in particular, to computer networks for creating and manipulating information containers with dynamic interactive registers in a computer, media or publishing network, in order to manufacture information on, upgrade the utility of, and develop intelligence in, a computer network by offering the means to create and manipulate information containers with dynamic registers.

2. Description of the Related Art

In the present day, querying and usage of information resources on a computer network is accomplished by individuals directing a search effort by submitting key words or phrases to be compared to those key words or phrases contained in the content or description of that information resource, with indices and contents residing in a fixed location unchanging except by human input. Similarly, the class of storage medium upon which information resides, it class and subclass organizational structures, and its routes of access all remain fundamentally unaltered by ongoing user queries and usage. Only the direct and intended intervention of the owner of the information content or computer hosting site changes these parameters, normally accomplished manually by programmers or systems operators at their own discretion or the discretion of the site owner.

There exists currently in the art a limited means of interfacing a computer user with the information available on computer networks such as the world wide web. Primarily, these means are search engines. Search engines query thousands or tens of thousands of index pages per second to suggest the location of information while the user waits. While factual information can be accessed, the more complex, particular or subtle the inquiry, the more branches and sub-branches need to be explored in a time consuming fashion in order to have any chance of success. Further, there are no such automatic devices that reconstruct the information into more useful groupings or makes it more accessible according to factors attached to the content by the content creator such as the space or time relevancy of its content, or factors attached to the content by the system's compilation and analysis of the accumulated biography of that specific content's readership.

The utility of wide area and public computer networks is thus greatly limited by the static information model and infrastructure upon which those networks operate.

One problem is that on a wide area or public network, specific content such as a document remains inert, except by the direct intervention of users, and is modified neither by patterns or history of usage on the network, or the existence of other content on the network.

Another problem is that content does not reside in an information infrastructure conducive to reconstruction by expert rule-based, fuzzy logic, or artificial intelligence based systems. Neither the intelligence of other information users nor the expert intelligence of an observant network computer system can be utilized in constructing, or re-constructing information resources. Where content resides in a fixed location and structure, "information" becomes something defined by the mind of the information provider rather than the mind of the information user, where the actual construction and utility of information exists. Information remains, like raw ore, in an unrefined state.

Another problem is that the class of storage medium upon which data resides cannot be system or user managed and altered according to the actual recorded and analyzed hierarchically graded usage of any given information resource residing on that storage medium except by statistical analysis of universal, undefined "hits" or visits to that page or site.

Another problem is that information resource groupings remain fixed on the given storage medium location according to the original installation by the resource author, not altered according to the actual recorded and analyzed hierarchically graded usage of that given information resource. Content itself remains inert, with no possibility of evolution.

A further problem with the prior art is that neither the search templates generated by those more knowledgeable in a given field of inquiry, nor the search strategies historically determined to be successful, or system-constructed according to analyses of search strategies historically determined to be successful, are available to inquiring users. A search template is here defined as one or more text phrases, graphics, video or audio bits, alone or in any defined outline or relational format designed to accomplish an inquiry. Internet or wide area network search may return dozens of briefs to a keyword or key phrase inquiry sometimes requiring the

time-consuming examination of multiple information resources or locations, with no historical relation to the success of any given search strategy.

A further problem is that there is limited means to add to, subtract from, or alter the information content of documents, databases, or sites without communicating with the owners or operators of those information resources, e.g., contacting, obtaining permission, negotiating and manually altering, adding or subtracting content. Additionally, once so altered, there is not a means to derive a proportionate value, and thereby a proportionate royalty as the information is used.

A final problem is that the physical residence of a body of data or its cyberspace location may not serve its largest body of users in the most expedient manner of access. Neither the expert intelligence of other information users nor the expert intelligence of an observant computer system is presently utilized by inherent network intelligence to analyze, re-design and construct access routes to information medium except by statistical analysis of universal, undefined "hits" or visits to that page or site.

Therefore, there is a need for a system and methods for creating and manipulating information containers with dynamic interactive registers defining more comprehensive information about contained content in a computer, media or publishing network, in order to manufacture information on, upgrade the utility of, and develop intelligence in, a computer network by providing a searching user the means to utilize the searches of other users or the historically determined and compiled searches of the system, a means to containerize information with multiple registers governing the interaction of that container, a means to re-classify the storage medium and location of information resources resident on the network, a means to allow the reconstruction of content into more useful formations, and a means to reconstruct the access routes to that information.

SUMMARY OF THE INVENTION

The present invention is a system and methods for manufacturing information on, upgrading the utility of, and developing intelligence in, a computer or digital network, local, wide area, public, corporate, or digital-based, supported, or enhanced physical media form or public or published media, or other by offering the means to create and manipulate information containers with dynamic registers.

The system of the present invention comprises an input device, an output device, a processor, a memory unit, a data storage device, and a means of communicating with other computers, network of computers, or digital-based, supported or enhanced physical media forms or public or published media. These components are preferably coupled by a bus and configured for multi-media presentation, but may also be distributed throughout a network according to the requirements of highest and best use.

The memory unit advantageously includes an information container made interactive with dynamic registers, a container editor, a search interface, a search engine, a search engine editor, system-wide hierarchical container gateways interacting with dynamic container registers, a gateway editor, a register editor, a data collection means with editor, a data reporting means with editor, an analysis engine with editor, an executing engine with editor, databases, and a means of communicating with other computers as above. These components may reside in a distributed fashion in any configuration on multiple computer systems or networks.

The present invention advantageously provides a container editor for creating containers, containerizing storing information in containers and defining and altering container registers. A container is an interactive nestable logical domain configurable as both subset and superset, including a minimum set of attributes coded into dynamic interactive evolving registers, containing any information component, digital code, file, search string, set, database, network, event or process, and maintaining a unique network-wide lifelong identity.

The container editor allows the authoring user to create containers and encapsulate any information component in a container with registers, establishing a unique network lifelong identity, characteristics, and parameters and rules of interaction. The authoring user defines and sets the register with a starting counter and/or mathematical description by utilizing menus and simple graphing tools or other tools appropriate to that particular register. The registers determine the interaction of that container with other containers, system components, system gateways, events and processes on the computer network.

Containers and registers, upon creation, may be universal or class-specific. The editor provides the means to create system-defined registers as well as the means to create other registers. The editor enables the register values to be set by the user or by the system, in which case the register value may be fixed or alterable by the user upon creation. Register values are

evolving or non-evolving for the duration of the life of the container on the system. Evolving registers may change through time, space, interaction, system history and other means.

System-defined registers comprise: (1) an historical container register, logging the history of the interaction of that container with other containers, events and processes on the network, (2) an historical system register, logging the history of pertinent critical and processes on the network, (3) a point register accumulating points based upon a hierarchically rated history of usage, (4) an identity register maintaining a unique network wide identification and access location for a given container, (5) a brokerage register maintaining a record of ownership percentage and economic values, and others.

The present invention also includes user-defined registers. User defined registers may be created wholly by the user and assigned a starting value, or simply assigned value by the user when that register is pre-existent in the system or acquired from another user, and then appended to any information container, or detached from any container.

Exemplary user-defined registers comprise (1) a report register, setting trigger levels for report sequences, content determination and delivery target, (2) a triple time register, consisting of a range, map, graph, list, curve or other representation designating time relevance, actively, assigning the time characteristics by which that container will act upon another container or process, passively, assigning the time characteristics by which that container be acted upon by another container or process, and neutrally, assigning the time characteristics by which that container will interact with another container or process, (3) a triple space register, consisting of a range, map, graph, list, curve or other representation designating the domain and determinants of space relevance, actively, assigning the space characteristics by which that content will act upon another container or process, passively, assigning the space, characteristics by which that content will be acted upon by another container or process, and neutrally, assigning the space characteristics by which that container will interact with another container or process, (4) a domain of influence register, determining the set, class and range of containers upon which that container will act, (5) a domain of receptivity register, determining the set, class and range of containers allowed to act upon that container, (6) a domain of neutrality register, determining the set, class and range of containers with which that container will interact, (7) a domain of containment register, determining the set, class and range of containers which that container may logically encompass, (8) a domain of inclusion register, determining the set, class and range of containers by which that container might be encompassed, (9) an ownership register, recording the original ownership of that containers, (10) a proportionate ownership register, determining the proportionate ownership of that containers, (11) a creator profile register, describing the creator or creators of that container, (12) an ownership address register, maintaining the address of the creator or creators of that container, (13) a value register, assigning a monetary or credit value to that container, and (14) other registers created by users or the system.

Containers are nestable and configurable as both subset and superset and may be designated hierarchically according to inclusive range, such as image component, image, image file, image collection, image database, or if text, text fragment, sentence, paragraph, page, document, document collection, document, database, document library, or any arrangement wherein containers are defined as increasingly inclusive sets of sets of digital components.

The present invention also includes, structurally integrated into each container, or strategically placed within a network at container transit points, unique gateways, nestable in a hierarchical or set and class network scheme. Gateways gather and store container register information according to system-defined, system-generated, or user determined rules as containers exit and enter one another, governing how containers system processes or system components interact within the domain of that container, or after exiting and entering that container, and governing how containers, system components and system processes interact with that unique gateway, including how data collection and reporting is managed at that gateway. The gateways record the register information of internally nested sub and superset containers, transient containers and search templates, including the grade of access requested, and, acting as an agent of an analysis engine and execution engine, govern the traffic and interaction of those containers and searches with the information resource of which they are the gateway and other gateways. The gateways' record of internally nested and transient container registers, and its own interaction with those containers, is made available, according to a rules-based determination, to the process of the analysis engine by the data collection and/or data reporting means.

The present invention also includes a means of data storage at any given gateway.

The present invention also includes a data collection means, residing anywhere on the network, or located at one or more hierarchical levels of nestable container gateways for gathering information from other gateways and analysis engines according to system, system-generated or user determined rules. The data collection means manages the gathering of data regarding network-wide user choices, usage and information about information, by collecting it from container and gateway registers as those containers and gateways pass through one another. Such statistics as frequency, pattern, and range of time, space and logical class is collected as directed by the analysis engine, and made that data available to the analysis engine by advancing it directly to the analysis engine, or incrementally, to the next greater hierarchically inclusive collection level. The rules of data collection may be manually set or altered by the system manager, or set by the system and altered by the system in its evolutionary capacity.

The present invention also includes a data reporting means, located at one or more hierarchical levels of nestable container gateways for submitting information to other gateways and analysis engines according to system, system-generated or user determined rules. The data reporting means manages the sending of data from the registers, gateways and search templates in a frequency, pattern, and range of time, space and logical class as directed by the analysis engine, and makes that data available to the analysis engine by advancing it directly to the analysis engine, or incrementally to the next greater hierarchically inclusive reporting level. The rules of data collection may be manually set or altered by the system manager, or set by the system and altered by the system in its evolutionary capacity. The data reporting means may be established to work in concert, in redundancy, or in contiguous or interwoven threads of hierarchically nested containers.

The present invention also includes an analysis engine that receives, reports and collects information regarding the interaction of user searches with gateways and container registers, as well as container registers with other container registers, and container registers with gateways. The analysis engine analyzes the information submitted by the gateways and instructs the execution engine to create new information containers, content assemblages, storage schemes, access routes, search templates, and gateway instructions. The analysis engine includes an editor that provides a system manager with a means of editing the operating principles of that engine, governing data reporting, data collection, search template loading, gateway instructions, and other.

The present invention also includes an execution engine, fulfilling the instructions of the analysis engine, to create new information containers, content sun and superset assemblages, storage schemes, access routes, search templates, and gateway instructions. The execution engine includes an editor that provides a system manager with a means of editing the operating principles of that engine, governing data reporting, data collection, search template loading, gateway instructions, and other.

The present invention also includes a search interface or browser. The search interface provides a means for a searching user to submit, record and access search streams or phrases generated historically by himself, other users, or the system. Search streams or phrases of other users are those that have been historically determined by the system to have the highest probability of utility to the searching user. Search streams or phrases generated by the system are those that have been constructed by the system through the analysis engine based upon the same criteria.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a first and preferred embodiment of a system constructed according to the present invention.

FIG. **2** A is block diagram of a preferred embodiment of the memory unit.

FIG. **2 B** is an exemplary embodiment of a computer network showing computer servers, personal computers, workstations, Internet, Wide Area Networks, Intranets in relationship with containers and gateways.

FIG. **2B1** is an exemplary embodiment of a computer network showing computer servers, personal computers, workstations, Internet, Wide Area Networks, Intranets in relationship with containers and gateways and exemplary locations of gateway storage in proximity to one or more of the various sites.

FIGS. **2C** through **2H** are exemplary embodiments in block diagram form of computer network components showing a possible placement of nested containers, computer servers, gateways, and the software components named in Fig. 2 A on a network.

FIG. **3A** is a graphical representation for one embodiment of a container having a plurality of containers nested within that container.

FIG. **3C** is a drawing showing elements that might be logically encapsulated by a container. FIG. **4** is a drawing of an information container showing a gateway and registers logically

encapsulating containerized elements.

FIG. **5** is a flowchart showing a preferred method for the containerization process and container editor operating on the communication device.

FIG. **6** is a flowchart showing a preferred method for searching for containers within a node.

FIG. **7** is a flowchart further showing a preferred method for searching for containers over one or more gateways.

FIG. **8** is a flowchart showing a method for performing the data collection and reporting on containers.

FIG. **9** is a flowchart showing the operation of the analysis engine.

FIG. **10** is a flowchart showing the operation of the execution engine.

FIG. **11** is a flowchart showing the operation of the gateway editor.

FIG. **12** is a flowchart showing the operation of the gateway process.

FIG. **13A** is a drawing showing an example of nested containers, gateways, registers, analysis engines and an execution engine prior to container reconstruction as depicted in 13 B, 13 C and 13 D.

FIG. **13B** is a drawing showing the reconstructed nested containers of Figure 13A.

FIG. **13C** is a drawing showing further reconstruction of nested containers, with a container relocated to reside within another container.

FIG. **13D** is a drawing showing a flowchart of the reconstruction process

FIG. **14** is a drawing showing the screen interface of the container editor.

FIG. **15** is a drawing showing the screen interface of the gateway editor.

FIG. **16** is a drawing showing the screen interface of the search interface.

FIG. **17** is a drawing of a generic application program showing a drop-down menu link, and a button link to the containerization process or container editor.

DESCRIPTION OF THE PREFERRED EMBODIMENT

THE SYSTEM

Referring now to FIG. 1, a preferred embodiment of a system **10** for creating and manipulating information containers with dynamic interactive registers in a computer, media, or publishing network **201** in order to manufacture information on, upgrade the utility of, and develop intelligence in that network **201**, is shown. The system **10** preferably comprises an input device **24**, an output device **16**, a processor **18**, a memory unit **22**, a data storage device **20**, and a communication device **26** operating on a network **201**. The input device **24**, an output device **16**, a processor **18**, a memory unit **22**, a data storage device **20**, are preferably coupled together by a bus **12** in a von Neumann architecture. Those skilled in the art will realize that these components **24**, **16**, **18**, **22**, **20**, and **26** may be coupled together according to various other computer architectures including any physical distribution of components linked together by the communication device **26** without departing from the spirit or scope of the present invention, and may be infinitely nested or chained, both as computer systems within a network **202**, and as networks within networks **201**.

The output device **16** preferably comprises a computer monitor for displaying high-resolution graphics and speakers for outputting high fidelity audio signals. The output device **16** is used to display various user interfaces **110**, **125**, **210**, **300**, **510**, **610**, **710**, as will be described below, for searching for and containerizing information, and editing the container gateways, containers, container registers, the data reporting means and the data collection means, and the search, analysis and execution engines. The author uses the input device **24** to manipulate icons, text, charts or graphs, or to select objects or text, in the process of packaging, searching or editing in a conventional manner such as in the Macintosh of Windows operating systems.

The processor **18** preferably executes programmed instruction steps, generates commands, stores data and analyzes data configurations according to programmed instruction steps that are stored in the memory unit **22** and in the data storage device **20**. The processor **22** is preferably a microprocessor such as the Motorola 680(x)0, the Intel 80(x)86 or Pentium, Pentium II, and successors, or processors made by AMD, or Cyrix CPU of the any class.

The memory unit **22** is preferably a predetermined amount of dynamic random access memory, a read-only memory, or both. The memory unit **22** stores data, operating systems, and programmed instructions steps, and manages the operations of all hardware and software components in the system **10** and on the network **201**, utilizing the communication device **26** whenever necessary or expeditious to link multiple computer systems **202** within the network **201**.

The data storage device **20** is preferably a disk storage device for storing data and programmed instruction steps. In the exemplary embodiment, the data storage device **20** is a hard disk drive. Historical recordings of network usage are stored on distributed and centralized data storage devices **20**.

The preferred embodiment of the input device **24** comprises a keyboard, microphone, and mouse type controller. Data and commands to the system **10** are input through the input device **24**.

The present invention also includes a communication device **26**. The communication device **26** underlies and sustains the operations of, referring now also to Fig **2** the analysis **400** and execution **500** engines, the data reporting **600** and collection **700** means, the container editor **110**, the search interface **300**, and the search engine **320**, providing the means to search, access, move, copy, utilize or otherwise perform operations with and on data. The communication device **26** utilizes one or more of the following technologies: modem, infrared, microwave, laser, photons, electrons, wave phenomena, cellular carrier, satellite, laser, router hub, direct cabling, physical transport, radio, broadcast or cable TV or other to communicate with other computers, digital-supported television, computer networks, or digital-based or supported public or published media, or physical media forms, on any a local, wide area, public, or any computer-based computer supported, or computer interfaced network, including but not limited to the Internet. It also allows for the functioning and distribution of any container **100** or container component herein described to reside anywhere on any computer system in any configuration on that local, wide area, public, or corporate computer-based or computer related network, or digital-based or supported media form.

Referring now to Figure **2** A, a preferred embodiment of the memory unit **22** is shown. The memory unit includes: an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510**, a search interface **300**, search engine **320**, analysis engine **400**, execution engine **500**, a data reporting module, **600**, a data reporting editor **610**, a data collection module **700**, a data collection editor **710**, screen interfaces (GUI's) **936**, menu or access buttons from generic computer programs **937**, and databases **900**, all residing in memory optimized between a data storage means **20** such as magnetic, optical, laser, or other fixed storage, and a memory means **22** such as RAM. The memory unit **22** functions by operating on communications network **12** with a communication device **26** on multiple computer systems **202** within the network **201**. These components will be described first briefly in the following paragraphs, then in more detail with reference to Figures 3 A through 17.

Those skilled in the art will realize that these components might also be stored in contiguous blocks of memory, and that software components or portions thereof may reside in the memory unit **22** or the data storage means **20**.

The present invention includes information containers **100** as noted above. The information container **100** is a logically defined data enclosure which encapsulates any element or digital segment (text, graphic, photograph, audio, video, or other), or set of digital segments, or referring now to FIG. **3 C,** any system component or process, or other containers or sets of containers. A container **100** at minimum includes in its construction a logically encapsulated portion of cyberspace, a register and a gateway. A container **100** at minimum encapsulates a single digital bit, a single natural number or the logical description of another container, and at maximum all defined cyberspace, existing, growing and to be discovered, including but not limited to all containers, defined and to be defined in cyberspace. A container **100** contains the code to enable it to interact with the components enumerated in 2 A, and to reconstruct itself internally and manage itself on the network **201**.

The container **100** also includes container registers **120**. Container registers **120** are interactive dynamic values appended to the logical enclosure of an information container **100**, and serve to govern the interaction of that container **100** with other containers **100**, container gateways **200** and the system **10**, and to record the historical interaction of that container **100** on the system **10**. Container registers **120** may be values alone or contain code to establish certain parameters in interaction with other containers **100** or gateways **200**.

The present invention also includes container gateways **200**. Container gateways **200** are logically defined gateways residing both on containers **100** and independently in the system 10. Gateways 200 govern the interactions of containers 100 within their domain, and alter the registers 120 of transiting containers 100 upon ingress and egress.

The present invention also includes container gateway storage 205 to hold the data collected from registers 120 of transient containers 100 in order to make it available to the data collection means 700 and the data reporting means 600, and to store the rules governing the operations of its particular gateway 200, governing transiting containers upon ingress and egress, and governing the interactive behavior of containers 100 within the container 100 to which that gateway 200 is attached. Gateway storage 205 may be located on gateways 200 themselves, containers 100 or anywhere on the network 202, 201, including but not limited to Internet, Intranet, LAN, WAN, according to best analysis and use.

The memory unit 22 also includes an execution engine 500 to perform the functions on the system 10 as directed by the analysis engine after its analysis of data from the data reporting means 600, the data collection means 700, and the search interface 300.

The memory unit 22 also includes a search interface 300, by which the user enters, selects or edits search phrases or digital strings to be used by the search engine 320 to locate containers 100.

The memory unit 22 also includes an analysis engine 400 which performs rules based or other analysis upon the data collected from the search interface 300 and the data collection 700 and data reporting 600 means.

The memory unit 22 also includes a data reporting means 600, by which means the information collected by gateways 200 from transient containers 100 is sent to the analysis engine 400.

The memory unit 22 also includes a data collection means 700, by which means the analysis engine 400 gathers the information collected by gateways 200 from transient containers 100.

The memory unit 22 also includes a container editor 110 for creating, selecting, acquiring, modifying and appending registers 120 and gateways 200 to containers 100, for creating, selecting, acquiring, and modifying containers, and for selecting content 01 to encapsulate.

The memory unit 22 also includes a register editor 125, for creating, selecting, acquiring and modifying container registers 120 and establishing and adjusting the values therein.

The memory unit **22** also includes a gateway editor **210**, by which means the user determines the rules governing the interaction of a given gateway **210** with the registers **120** of transient containers **100**, governing transiting containers upon ingress and egress, and governing the interactive behavior of containers within the container to which that gateway is attached.

The memory unit **22** also includes databases **900**, by which means the analysis engine **400**, the execution engine **500**, the gateways **100**, the editors **110**, **125**, **210**, **510**, **610**, **710**, and the search interface **300**, store information for later use.

The memory unit **22** present invention also includes a search engine **320** by which means the user is able to locate containers **100** and, referring now to Fig. 4, containerized elements **01**.

The memory unit **22** present invention also includes an engine editor **510**, by which means the user establishes the rules and operating procedures for the analysis engine **400** and the execution engine **500**.

The memory unit **22** present invention also includes a reporting means editor **610**, by which means the user establishes the rules and schedule under which the information collected by gateways **200** from transient containers **100** will be sent to the analysis engine **400**.

The memory unit **22** present invention also includes a collection means editor **710**, by which means the user establishes the rules and schedule under which the analysis engine **400** will gathers the information collected by gateways **200** from transient containers **100**.

The memory unit **22** present invention also includes screen interfaces (GUI's) **936**, specifically designed to simplify and enhance the operations of the container editor **110**, the gateway editor **210**, and the search interface **300**.

The present invention also includes a menu or button access **937**, by which a user utilizing any generic computer program may access the system **10** or the container editor **110** from a menu selection(s) or button(s) within that program.

The present invention also includes a computer, media or publishing network **201**, comprising computers, digital devices and digital media **202** and a communication device **26**, within which the components enumerated in Fig. 2 A interact, compiling, analyzing, and altering containers **100** and the network **201**according to information gathered from container registers **120**.

The memory unit **22** also includes one or more computers **202**, by which means the components of Fig **1** sustain the operations described in Fig. **2 A.**

The memory unit **22** also includes flat or relational databases **900**, used where, and as required. Databases are used to store search phrases, search templates, system history for the analysis engine and execution engine, container levels and container, sites and digital elements, or any and all storage required to operate the system.

Referring now to FIG. **2 B,** a drawing of a computer network **201** as a system **10,** showing a possible placement of nested containers **100**, computer servers, gateways **200**, on the sites described below. (Note: Fig. 2 B utilizes in parts the same numbering scheme as Fig. 13 A, 13 B, 13 C, 13 D and as Fig. 2 A.) In FIG. **2 B** various exemplary sites are shown, any or all of which might interact dynamically within the system. Site 1 shows a single workstation with a container and gateway connected to an Intranet. .(Individual containers may be a floppy or CD-Rom to be downloaded or inserted.) Site 2 shows a server with a gateway in relationship to various containers.. Site 3 shows an Internet web page with a container residing on it. Site 4 shows a personal computer with containers and a gateway connected to the Internet. Site 5 shows a configuration of multiple servers and containers on a Wide Area Network.. Site 6 shows a workstations with a gateway and containers within a container connected to a Wide Area Network. Site 7 shows an independent gateway, capable of acting as a data collection and data reporting site as it gathers data from the registers of transiting containers, and as an agent of the execution engine as it alters the registers of transient containers. A container **100** contains the code to enable it to interact with the components enumerated in **2A,** and to reconstruct itself internally and manage itself on the network **201.** The code resides in and with the container in its registers and gateway definitions and controls. Additional system code resides in all sites to manage the individual and collective operation and oversight of the components enumerated in **2A**, with the specific components distributed amongst the sites according to the requirements of optimization.

Referring now to Fig. **2 B 1** various exemplary sites are shown as described above in Fig. **2 B**, with the addition of possible location of one or more gateway storage **205** locations.

Referring now to Figures **2 C** through **2 H**, various exemplary sites with one or more of the logical components of the system 10 in relationship are shown. Site 1 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510**, a search interface **300**, search engine **320**, analysis engine **400**, execution engine **500**, a data reporting means **600**, a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26**.

Site 2 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, system-wide hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, engine editors **510**, search engine **320**, analysis engine **400**, execution engine **500**, a data reporting means **600**, a data reporting means editor **610**, a data collection means **700**, a data collection means editor **710**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26**.

Site 3 comprises an interactive information container 100, a container editor **110**, container registers **120**, a container register editor **125**, hierarchical container gateways 200, gateway storage 205, gateway editors **210**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26**.

Site 4 comprises an interactive information container **100**, a container editor **110**, container registers **120**, a container register editor **125**, hierarchical container gateways **200**, gateway storage **205**, gateway editors **210**, a search interface **300**, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26**.

Site 5 comprises an interactive information container **100**, container registers **120**, a container register editor **125**, hierarchical container gateways **200**, gateway storage **205**, and databases **900**, all residing on data storage means **20**, accessed and utilized by non-resident memory unit **22**, operating on communications network **12** with a communication device **26**.

Site 6 includes an independent analysis engine **400**, execution engine **500**, data collection means **700**, and data reporting means **600** gateway editors **210**, engine editors **510**, a data reporting means editor **610**, a data collection means **700**, a data collection means editor

710, and databases **900**, all residing on data storage means **20**, utilizing the memory unit to function **22**, operating on communications network **12** with a communication device **26.**

Referring now to FIG. **3 A** and FIG. **3 B**, a block diagram of several nested information containers is shown, including examples of elements, e.g., code **1100**, text **1200**, audio **1300**, video **1400**, photograph **1500**, graphic images **1600**, and examples of possible container level classifications in increasing size, e.g., element **10900000**, document **10800000**, database **10700000**, warehouse **10600000**, domain **10500000**,and continuing increasingly larger on Fig **3 (B)**, subject **10400000**, field **10300000**, master field **10200000**, species **10100000**. Containers may be infinitely nested and assigned any class, super class or sub class scheme and description by the creator of the container to govern nesting within that container. In addition to digital elements, containers may also include system process and components, including containerization itself.

Referring now to FIG. **3 C**, a block diagram of an information container system is shown, listing, without any relationship indicated, some of the possible system components and processes, or sets thereof, that may be encapsulated as elements **01** in an information container **100.** An information container 100 may include one or more of the following: any unique, container **100**, gateway **200**, output device **16**, input device **24**, output device process **160**, input device process **240**, data storage device **20**, data storage device process **2000**, processor **18**, bus **12**, content **01**, search process **02**, interface **04**, memory unit **22**, communication device **26**, search interface **300**, search process **98**, network **201**, class of device, process or content **999**, class of process at any unique class of device **990**, process at any unique device **99**, editor **110**, **125**, **210**, **510**, **610**, **710**, engine **320**, **400**, **500**, containerization process **1098**, or process **08**.

Any container may include (n) other containers, to infinity. The use of value evolving container registers **120** in conjunction with gateways **200**, data reporting modules **600**, data collection modules **700**, the analysis engine **400**, and the execution engine **500** provides the information container **100** with extensive knowledge of the use, operation of its internal contents, prior to, during and after those contents' residence within that container **100**, and extensive knowledge of the use, operation and contents of the system **10** external to itself, and allows the container **100** to establish and evolve its own identity and course of interaction on the system **10**. Further, containers **100**, as logical enclosures, can exist and operate independent of their digital contents, whether encapsulating audio, video, text, graphic, or other.

Referring now to FIG. **4**, a block diagram of an information container **100** is shown. The information container **100** is a logically defined data enclosure which encapsulates any element, digital segment (text, graphic, photograph, audio, video, or other), set of digital segments as described above with reference to FIG. **3 (C)**, any system component or process, or other containers or sets of containers. The container **100** comprises the containerized elements **01**, registers **120** and a gateway **200**.

Registers **120** appended to an information container **110** are unique in that they operate independently of the encapsulated contents, providing rules of interaction, history of interaction, identity and interactive life to that container **100** through the duration of its existence on a network **201**, without requiring reference to, or interaction with, its specific contents. They enable a container **100** to establish an identity independent of its contents. Additionally, registers **120** are unique in that their internal values evolve through interaction with other containers **100**, gateways **200**, the analysis engine **400**, the execution engine **500**, and the choices made by the users in the search interface **300**, the container editor **110**, the register editor **125**, the gateway editor **210**, the engine editor **510**. Registers **120** are also unique in that they can interact with any register of a similar definition on any container **100** residing on the network **201**, independent of that container's contents. Registers **120**, once constructed, may be copied and appended to other containers **100** with their internal values reset, to form new containers. Register values, when collected at gateways **200** and made available to the analysis engine **400** through the data collection means **700** and the data reporting means **600**, provide an entirely new layer of network observation and analysis and operational control through the execution engine **500**. Registers **120** accomplish not only a real time information about information system, but also a real time information about information usage on a network. Further, because the user base of a network determines usage, the system **10**, in gathering information about information usage, is observing the choices of the human mind. When these choices are submitted to the analysis of a rules-based or other analysis engine **400**, the system **10** becomes capable of becoming progressively more responsive to the need of the user base, in effect, learning to become more useful by utilizing the execution engine **500** to create system-wide changes by altering the rules of gateway **200** interaction and thereby altering the registers **120** of transient containers **100** and establishing a complete evolutionary cycle of enhanced utility.

Further, in establishing the pre-defined registers as described in the following four paragraphs, the following unique aspects of information about information are utilized for the first time: 1) the dynamic governance of information according to its utility through time, in active, passive and neutral aspects, as explained below; 2) the dynamic governance of information according to its utility through space in active, passive and neutral aspects, as explained below; 3) the dynamic governance of information according to its ownership, as explained below; 4) the dynamic governance of information according to its unique history of interaction as an identity on a network, as explained below; 5) the dynamic governance of information according to the history of the system on which it exists, as explained below; 6) the dynamic governance of information according to established rules of interaction, in active, passive and neutral aspects, as explained below; 7) the dynamic governance of information according to the profile of its creator, as explained below; 8) the dynamic governance of information according to the value established by its ongoing usage, as explained below; 9) the dynamic governance of information according to it distributed ownership, as explained below; 10) the dynamic governance of information according to what class of information it might be incorporated into, and according to what class of information container it might incorporate, as explained below; 11) the dynamic governance of information according to self-reporting, as explained below.

Referring now to Fig **4**, registers **120** may be (1) pre-defined, (2) created by the user or acquired by the user, or (3) system-defined or system-created. Pre-defined registers **120** are those immediately available for selection by the user within a given container editor as part of that container editor, in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. Registers **120** created by the user are those conceived and created by a specific user or user group and made immediately available for selection by the user or user group in conjunction with any of a wide number of container editors, in order that the user may append any of those registers **120** to a container **100** and define values for those registers **120** where required. Registers **120** acquired by the user are those registers existing network-wide **201**, created by the user base, that might be located and acquired by the user in order that the user may append any of those registers

120 to a container **100** and define values for those registers **120** where required. System-defined registers are those registers whose values are set and/or controlled by the system **10**. System-created registers are those registers created by the system **10**.

Registers **120** are user or user-base created or system-created values or ranges made available by the system **10** to attach to a unique container, and hold system-set, user-set, or system-evolved values. Values may be numeric, may describe domains of time or space, or may provide information about the container **100**, the user, or the system **10**. Registers **120** may be active, passive or interactive and may evolve with system use. Pre-defined registers include, but are not limited to, system history **110000**, container history **101000**, active time **102000**, passive time **103000**, neutral time **104000**, active space **111000**, passive space **112000**, neutral space **113000**, containment **105000**, inclusion **106000**, identity **114000**, value **115000**, ownership **107000**, ownership addresses **116000**, proportionate ownership **117000**, creator profile **108000**, receptivity **118000**, influence **119000**, points **109000**, others **120000**, reporting **121000**, neutrality **122000**, acquire **123000**, create **124000**, content title **125000**, content key phrase(s) **126000**, and content description **127000**, security **12800**, and parent rules **129000**.

Pre-defined registers comprise an historical container register **101000**, logging the history of the interaction of that container **100** with other containers, events and processes on the network **201**, an historical system register **110000**, logging the history of pertinent critical and processes on the network, a point register **109000** accumulating points based upon a hierarchically rated history of usage, an identity register **114000** maintaining a unique network wide identification and access location for a given container specifying a unique time and place of origin and original residence, a proportionate ownership register **117000** maintaining a record of ownership percentage and economic values, and others **120000**.

User-defined registers include a report register **121000** setting trigger levels for report sequences, content determination and delivery target, three time registers, consisting of a range, map, graph, list, curve or other designating time relevance, **102000** assigning the time characteristics by which that container will act upon another container or process, **103000** assigning the time characteristics by which that container be acted upon by another container or process, and **104000** assigning the time characteristics by which that container will interact with another container or process, three space registers, consisting of a range, map, graph, list, curve or other designating the domain and determinants of space relevance, **111000** assigning the space characteristics by which that content will act upon another container or process, **112000** assigning the space, characteristics by which that content will be acted upon by another container or process, and **113000** assigning the space characteristics by which that container will interact with another container or process, a domain of influence register **119000**, determining the set, class and range of containers upon which that container will act, a domain of receptivity register **118000**, determining the set, class and range of containers allowed to act upon that container, a domain of neutrality register **122000**, determining the set, class and range of containers with which that container will interact, a domain of containment register **105000**, determining the set, class and range of containers which that container may logically encompass, a domain of inclusion **106000** register, determining the set, class and range of containers by which that container might be encapsulated, an ownership register **107000**, recording the original ownership of that containers, a creator profile register **108000**, describing the creator or creators of that container, an ownership address register **116000**, maintaining the address of the creator or creators of that container, a value register **115000**, assigning a monetary or credit value to that container, other registers **120000** created by users or the system, a reporting register **121000**, determining the content, scheduling and recipients of information about that container, a neutrality register **122000**, an acquire register **123000**, enabling the user to search and utilize other registers residing on the network, a create register **124000**, enabling the user to construct a new register, a content title register **125000**, naming the contents of the container, a content key register, **126000**, identifying the container contents with a key phrase generated by the user and/or the system based upon successful usage of that phrase in conjunction with the utilization of the information within that container **100**, a content description register **127000**, identifying the container contents with additional description, a security register **128000**, controlling container security, and a parent container register **129000**, storing the rules governing container interaction as dictated by the parent (encapsulating) container.

The container also includes a gateway **200** and gateway storage **205.**

Gateways **200** are logically defined passageways residing both on containers **100** and independently in the system **10**. Gateways **200** govern the interactions of containers **100** encapsulated within their domain by reading and storing register **120** information of containers entering and exiting that container **100**.

The present invention also includes container gateway storage **205**. Gateway storage **205** stores information regarding the residence, absence, transience, and alteration of encapsulated and encapsulating containers **100**, and their attached registers **120**, **holding** the data collected from registers **120** of transient containers **100** in order to make it available to the data collection means **700** and the data reporting means **600**, and storing the rules governing the operations of its particular gateway **200**.

Referring now to FIG. **5**, a flow chart of the preferred method for creating a container **100**.is shown.

Input is received from the user selecting a container level through use of a drop-down menu **10100**. A menu of all possible container classes within the subset and superset scheme of multiple hierarchically nested containers, i.e.; element, document, file, database, warehouse, domain, and more, is displayed on the output device **10200**. Input is received from the user selecting a class **10300**.

A graphic representation of a container in that class, with registers common to all containers as well as registers unique to its class is displayed **10301**.

Input is received from the user choosing to "create" **10400**, "edit" **10500**, or "locate" **10600**.

When the input of "create" **10400** is received from the user, a container template in that class appears **10410**. Input from the user is then received adding or selecting a register 10540 to append to that container template. When input is received from the user adding a register, a list of registers that might be added to that class of container is made available to select **10550**. Input is received from the user selecting a register **10560** and editing it **10570**. The menu returns to "add or select" **10540**.

If the input of "locate" **10600** is received from the user, the system prompts the user to enter the identity of the container or class of containers **10605**. The system locates the container(s) **10610**. Input is received from the user selecting a container **10620**. The system prompts the user for a security code for permission to access the container for template use, or to alter its registers, or to alter its content **10630**. . Input is received from the user entering a name and password providing access to one of the security levels **10640**. Input is received from the user editing the container accordingly by transition to step **10500** and performing the steps for editing.

If the input of "edit" **10500** is received, a list of containers available to edit at that level is shown **10510**. Input is received from the user selecting a container **10520**. That container appears, available to edit **10530**. Input is received from the user selecting "add" or "select" registers **10540** by the user clicking on the graphically depicted register, or from a drop down menu. Input is received from the user selecting the register to edit **10560**. Input is received from the user selecting "modify" or "delete" for that register **10565**. If input is received from the user to "delete," that register is severed from the container. If input is received from the user to "modify", the register editor **10570** screen appropriate to that register appears, i.e., an x-y type graph to define a curve of relevant active time, in which the user manipulates the x-y termini, scale and curve, or a global map in which Input is received from the user selecting the locale of active space, whether zip code, city, county, state, country, continent, plant or other. When input is received from the user saving the definition, the screen returns to the main container screen to make another selection available. . Input is received from the user defining as many registers as he chooses. One of the registers may be named "new register." Input is received from the user selecting the new register, and if chosen by the user, defining a wholly unique and new kind of register by the user entering input into the register editor **125.**

When the input is received from the user choosing to add a register, a list of registers that might be added to that class of container are made available to select **10550**. Input is received from the user selecting a register **10560** and editing it **10570**. The menu returns to "add or select" **10540**, and in turn to Input – Select Container.

Input may then be received from the user choosing to add, modify, or delete the container contents **10700**. Once the registers are defined, input is received from the user indicating completion and the interface reverts to the container editor. When input is received from the user choosing "select component" (to select the component to containerize) from the main menu bar **10700,** a window appears allowing the user to select any file, component, or other container. If for example, the user were creating a warehouse container, and wishes to incorporate several databases into that container, input would then be received from the user selecting "database." The program would prompt the user for the location (directory) of that database or container. If the requested selection is not containerized, input may then be received from the user choosing to containerize the element at that time, after which the program returns to "select component." Once input is received from the user defining the database location, the program logically encases the directory or directories in the defined container. The above procedure may be repeated as many times as desired to include multiple databases within a single container. While logical simplicity would dictate that all containers within a container be of the same subset, it would be possible for input to be received from the user choosing containers of any subset to include in the container. When input is received from the user choosing "finished," the container is created with a unique network identity, preferably through some combination of exact time and digital device serial number, or centralized numbering system, or other means. The container **100** contains all digital code, including data and program software from the selected items or containers.

Input may then be received from the user to publish the container **11100** at a user-identified or system suggested location **11200** to be selected **11400**.

Input is received from the user to "publish", from the main menu bar **11100**. Input is received from the user choosing to leave the container where it was created, move or copy it to another drive, directory, computer, or network the user designates, or select the location from location options offered by the system **11200**, or submit, or duplicate and submit, the container to the analysis engine **400** for intelligent inclusion in other containers, thus allowing the system to publish the container as instructed or choose the residence of the container **11400**.

If input is received from the user to choosing to "move," or "copy" a browse function allows the user to name the new location or browse a list of possible locations. If input is received from the user choosing to "submit," a browser function allows the user to name the analysis search engine **310** or browse a list of possible analyses engines. When input is received from the user choosing the residence of the container **11300**, the program restores the search interface screen.

Referring now to FIG. **6**, a flow chart of the method for searching for containers **100**.

When input is received from the user selecting "search interface" from the main title bar, the search interface screen appears. The user is given the choice of containerizing selected content or requesting that container levels be displayed **30100**. From a drop down menu another menu appears allowing input to be received from the user selecting the container level

30200. Input is received from the user selecting the container level (from the smallest component to the whole system) **30300.**

Input is received **30310** from the user selecting the phrases, containers or components, which then are re-submitted to the same process, until the input is received from the user selecting a specific site or container.

The search phrase, whether containerized or not, is submitted simultaneously to the search engine **30400** and the analysis engine **30500.**

The screen then reports in a selection menu, the number of applicable sites found by the search engine **30410**, the number of historically proven applicable sites found by the analysis engine **30410**, the number of historically proven applicable containers at the selected container level or any container level found by the analysis engine **30410**, and the number of historically proven new search phrases or digital segments found by the analysis engine **30320**. . Input is received from the user selecting one of the named sets above **30330**. If input is received from the user choosing the search engine, the search interface lists the applicable site titles with a brief description **30410**. If input is received from the user choosing the site list of the analysis, the search interface lists the applicable site titles with a brief description **30410**. . If input is received from the user choosing the container list of the analysis engine, the search interface lists the applicable container titles with a brief description **30410**. . If input is received from the user selecting a container **30420**, the system offers the means to view titles and descriptions of sub-containers at any chosen class level. . If input is received from the user choosing the phrase list of the analysis engine, the search interface lists the applicable phrases or digital segments with a brief description **30320**. The search and search result cycle repeats until input is received from the user choosing to go to an individual container or site.

Input is received from the user entering text or any digital string describing his search objectives into a text or search box. When input is received from the user submitting the search string, the system provides the option of containerizing the search through the container editor **110.** Once the search container **101** is created, the system restores the search interface **300** screen the user.

Input is received from the user selecting "search", "supported search" or "both" from another drop-down menu and from submitting the search. When input is received from the user selecting "search" 30310, the search phrase is submitted to the search engine **30400**, which searches both content and the appropriate container registers, as pre-indexed in the search engine, and returns a list of appropriate locations, components or containers. When input is received from the selecting "supported search", the search phrase is submitted to the analysis engine search support, which returns a list, in a drop-down menu, of search phrases or individual containers, for any and all container levels, used by other users or created by the system and known to be historically successful for the described effort and the described searching user, as per the results of the analysis search engine. Input is received from the user selecting a new search phrase or specific container from the drop down menu **30330**. When input is received from the user choosing a new search phrase, that phrase is also submitted to the analysis engine **30500** which returns a list of pre-compiled historically proven sites, components or containers associated with that search phrase **30320**. Input is received from the user choosing a selection **30420** and the system calls up that specific site, container or component. If input is received from the user selecting a specific site, container or component at any time during the search process, that element is called up by the system **30440**.

Input is received from the user choosing to containerize a search or select a container level in which to search **30100**. When input is received from the user choosing to containerize the search, the software moves to the container editor as described in Fig. 5, and then returns the user to the search interface screen. Input is received from the user selecting to search a specific container level or the whole network. The system shows the available levels **30200**. Input is received from the user selecting a container level **30300**, and entering the text or digital component comprising the search string **30310**. The system searches the containers **30400** while simultaneously submitting the search string to the analysis engine **30500**. While the system is accessing containers, sites or templates **30700**, the analysis engine **30500** inquires of the appropriate database **30600** to access historically successful containers, sites or search templates corresponding to the search request **30700**, which is then shown on another portion or option of the search interface, either as available containers or sites **30410** or as search template options **30320**. On one portion or option of the search interface screen the corresponding containers or sites are listed and/or previewed for selection **30410**. Input is received from the user selecting the container to access **30420**. The system accesses that container **30430** and shows it on the screen **30440** for user review. Input is received from the user selecting an operation, i.e., preview, read, purchase, move, copy, lease, in any composed schedule with operations assigned specific values **30460**, and the system obtains the specified result **30470**. The selection of the operation including any interaction with any uniquely defined container **100** is recorded **30800** by the container gateway (Fig. 2 A, **200**), stored in the gateway storage **205** and made available to the analysis engine (Fig. **9**) by the data collection and reporting means (Fig. **8**). Reporting and collection occurs on a regular basis according to user determined times or rules. The analysis engine compiles and analyzes selections according to various rules-based systems applicable to the particular container area of residence in cyberspace.

. Input is received from the user selecting the container or site **30410**, proceeding as described above, or selecting a search template **30330**, and editing it to re-enter the search **30310**. All operations on Fig. **6** utilize the communication device **26** whenever necessary or expeditious.

Referring now to FIG. **7**, a flow chart of the search process is shown. Steps in FIG. **7** repeated from FIG. **6** are given the same reference number as in FIG. **6** for convenience and ease of understanding. Fig. **7** commences with "SEARCH TRANSITS GATEWAY **32100**", continuing from Fig. **6**, "SYSTEM SEARCHES CONTAINERS **30400**". The submitted search **32100** transits the gateway **200**. The gateway **200** interacts with the container registers **32200**. The gateways **200** store the information downloaded from the registers **32300**, and the container registers are altered **32500**. The container registers **120** then interact with the registers **120** of the encapsulated search, which registers, and the values set within, have been constructed and appended to the search through the search interface **32600**. Values are exchanged and compared and operations performed under the rules governing both interacting containers **100**, and the rules governing the search container **100** and any gateway **200**. The search engine **320**, operating under the principles and means of search engines presently existing as described elsewhere, **then** provides to the search interface **32600** a list of containers **100** meeting the requirements of the search and its appended registers, as well as additional search options **32900**. The gateway **200** reports and makes available for collection to the analysis engine **400** the information obtained from the interaction **32400**. On a periodic basis defined by the user or a rules-based system, the analysis engine **400** (Fig. **9**) stores in databases **900**, analyzes and instructs the execution engine **500**, and the execution engine **500** executes changes in the system components as defined below. (Fig. **10**). All operations on Fig. **7** utilize the communication device **26** whenever necessary or expeditious.

On the remaining figures, shapes referring to other figures, to operations external to the scope of the present figures, or to the subject of the present drawing, are indicated with dashed lines, and are shown only to place the described operations in the context of continuous and continual operations external to the drawing.

Referring now to FIG. **8**, a flow chart of the preferred process for collecting and reporting information on containers is shown. The data reporting **600** and data collection **700** means utilizes subroutines within the analysis engines **400** and gateways **200** to submit and collect register information and sub level analysis to other analysis engines **400** or other gateways **200** of a higher (larger) logical set in a set pattern and frequency defined by the administrator.

Input is received from the user selecting "data reporting" **70100** from the "edit gateway" drop-down menu. Container levels are displayed **70200**. Input is received from the user selecting container level **70300.** A menu of all possible gateways **70320** and analysis engines **70330** residing on gateways on the above defined container class appears, depicted graphically as a tree of analysis engines and gateways at that container level. Input is received from the user selecting "source" from "source or destination." Input is received from the user **70400** selecting a container, containers, or class of container by clicking on the graphically depicted container(s) or container level on a display device. Input is received from the user **70410** selecting "destination" from "source or destination" Input is received from the user **70500** selecting an analysis engine, analysis engines, or class of analysis engine by clicking on the graphically depicted analysis engine(s) or analysis engine level on a display device. A time scheduler is displayed. Input is received from the user **70510** selecting the reporting frequency for the selected gateways to report data to the selected engines. The data from the gateways is thenceforth continuously moved or copied to the analysis engines by the system **10** utilizing the execution engine **500** according to the defined schedule, rules and pattern **70420, 70520**.

Input is received from the user selecting "choose container level" **70300** from the gateway editor drop-down menu. A menu **70320** appears listing the classes of containers on the system within the defined subset and superset scheme of multiple hierarchically nested containers, i.e.; element, document, file, database, warehouse, domain, appears. Input is received from the user selecting the class of containers. A graphic representation of that container level throughout the system appears. Input **70300** is received from the user selecting individual containers or all the containers in that class.

From the gateway editor drop-down menu input **70100** is received from the user selecting "data collecting" A menu of all possible gateways and analysis engines residing on gateways on the above defined container class appears, depicted graphically as a tree of analysis engines, and gateways at that container level. Input **70510** is received from the user selecting "source" from "source or destination." Input is received from the user selecting a container, containers, or class of container by clicking on the graphically depicted container(s) or container level. Input **70510** is received from the user selecting "destination" from "source or destination." Input **70510** is received from the user selecting an analysis engine, analysis engines, or class of analysis engine by clicking on the graphically depicted analysis engine(s) or analysis engine level. A time scheduler appears. Input **70510** is received from the user selecting the collecting frequency for the selected engines to collect data from the selected gateways. The data from the gateways is thenceforth continuously moved or copied to the analysis engines by the system **10** utilizing the execution engine **500** according to the defined schedule, rules and pattern.

The data collection **700** means, utilizing the communication device **26** and an execution engine **500**, comprises one or more subroutines or agents programmed to travel through the network collecting the accumulated data and analyses from selected analysis engines, gateways or selected subset level of analysis engines or gateways (as above) in a pattern and frequency defined by the gateway administrator at a given container level. Input **70510** is received from the user or administrator, defining the collection and reporting of data, thus controlling permission within his gateway, and being subject to permission levels defined by others beyond his gateway.

Input is received from the user or gateway administrator selecting collection or reporting **70100** and the system shows the container levels available **70200**. Input is received from the user selecting a container level **70300**. Input is received from the user selecting "gateway" **70400** or "engine" **70500**. The system shows gateways **70320** or engines **70330** associated with that level. Input is received from the user editing the reporting parameters associated with a gateway or a class of gateways **70410** or an engine or class of engines **70510**.

Input is received from the user selecting the collecting frequency for the chosen engines. When input is received from the user choosing to user save the definition, the screen returns to the main container screen, step **70100** to make another selection available. Input is received from the user choosing to repeat the cycle, choosing "destination" to describe the destination analysis engines and the data collecting frequency from those destination analysis engines. The data collection means **700** collects the accumulated gateway information in a pattern and frequency defined by the gateway administrator or user at a given container level.

The system utilizing the execution engine (see Fig. **10**) distributes the new parameters to the gateways **70420** or engines **70520** by the communication device **26**. Using the new parameters the gateways report to the analysis engines **70430** after, in some cases, conducting sub-analysis **70440,** or using sub-analysis **70440** to submit directly to specified gateways under certain conditions and parameters, and the analysis engines collect from the gateways **70530**. The analysis engine uploads, downloads and utilizes information to databases **900** to conducts its analysis.

The invention includes an analysis engine **400**. Through the data reporting **600** means and data collection **700** the analysis engine **400** receives data and sub-analysis from the search interface and the gateways. Data includes, for each gateway **200**, the frequency and grade of access, the description of the user accessing, the identity of the container **100** accessing, the register parameters, and the historically accumulated register data.

Referring now to FIG. **9**, a flow chart of the operation of the analysis engine **400** is shown. Analysis engines **400** may reside at any gateway or anywhere in the system 10. The analysis engine **400,** operating under its own programmed sequence, utilizing the communication device **26**, works, by means of programmed rules of logical, mathematical, statistical or other analysis upon gateway and register information, in continuous interaction with the search process **410** and the data collection and reporting process **420** to analyze, determine and compile instructions **40100** on container construction **40110** to containerize in an automated process **40115**, on container contents **40120** to move, copy or delete containers **40125**, on storage schemes **40130** to move or copy containers to new storage **40135**, on access routes **40140** to alter gateway pointers to sought information **40145**, on search templates **40150** to add, delete or change search phrases and the referenced objects indicated by those search phrases **40155** and on gateway instructions **40160** to alter gateway registers and pointers **40165**.

Thus, analyses might include, but are not limited to, the physical locus of the users accessing, the demographic classification of the users accessing, the access frequency for a given container, the range or curve of time relevance affecting a container, the range or region of space relevance affecting a container **100**, the number or number of a specific type of container **100** transiting a gateway **200**, the hierarchically graded usage of containers **100** or container contents **01** compared with the demographic of those users accessing the container, the hierarchically graded usage of containers **100** or container contents **01** compared with search phrases entered into the search interface **300**, the hierarchically graded usage of containers **100** or container contents **01** compared with search phrases entered into the search interface 300 compared with the demographic of the users accessing, the number of pertinent containers nested within a given container **100**. Once an analysis is accomplished, the result is compared to pre-programmed rules triggering instruction sets (such as moving a container to nest within another container).

Instructions are then sent to the execution engine **40200**, which utilizes the communication device **26** to execute the instructions derived from the analyses. These containerized instructions transit the gateways **40300** and are utilized in the gateway process (Fig. **12**)

Referring now to FIG. **10**, a flow chart of the operation of the execution engine is shown. The execution engine **400**, operating under its own programmed sequence in response to the instructions from the analysis engine **50100**, utilizing the communication device **26**, works in continuous process as its containerized execution instructions transit the gateways **50200** to create containers **50210** in an automated containerization process **50215**, alter container contents **50230** by moving or copying containers to new containers **50235**, to alter storage **50240** by moving or copying containers to new storage **50245**, to alter access routes **50250** by altering gateway pointers **50255**, to alter search templates **50260** by adding, changing and deleting search phrases and the referenced objects indicated by those search phrases **50265**, to alter gateway instructions **50270** by altering gateway registers and pointers **50275**. The execution works in a continuous loop with the gateway process **50300**, the data collection and reporting process **50400** and the analysis engine process **50300**.

The invention includes gateways **200**. Gateways may be placed and reside anywhere on the network where containers transit. Gateways also reside on any or all containers. The gateway reads and stores the chosen register information from transient containers entering or exiting its logical boundaries. The resident analysis search engine, if any, performs the specified level of analysis. Data and analysis is both held for the collection means according to the pattern and timing specified in the data reporting **600** editor and submitted according to the pattern and timing specified in the data collection means editor **700.**

The gateways are network-wide, hierarchical, and nestable, and reside with a container encompassing any component, digital code, file, search string, set, database, network, event or process and maintaining a unique lifelong network wide identity and unique in all the universe historical identity, or may be strategically placed at such container transit points to gather and store register information attached to any such container, according to system-defined, system-generated, or user determined rules residing in its registers defining the behavior of those containers and components as they exit and enter one another, or interact with one another or any system process or system component within the logical domain of that container, or after exiting and entering that container, or defining how they interact with that unique gateway.

Gateway's registers comprise both system-defined and user-defined registers, alterable by author, duration, location, network-wide history, individual container history and/or interaction with other containers, gateways, networks or media, and evolve according to that gateway's history on a computer network, or according to the network history of events and processes, or according to that information component's interaction with other information containers, components, system components, network events or processes.

Referring now to FIG. **11,** a flow chart of the gateway editor is shown. From the main title bar input is received from the user selecting "containerize" or "gateway level" **20100.** When input is received from the user selecting "containerize" the system enters the container editor process **110.** When input is received from the user selecting "gateway," the system shows the gateway levels available **20200.** A menu of all possible gateways within the subset and superset scheme of defined multiple hierarchically nested gateways appears. Input is received from the user selecting the gateway level **20300.** The system searches the gateways **20500** to locate the available gateway templates **20700** and the available gateways **20600.** Input is received from the user selecting the gateway **20610** or gateway level template **20720.**

The system goes to the gateway **20620** or to the template **20720**. A graphic representation of the chosen gateway **20630** or template **20730** appears. Input is received from the user to edit **20640** or create a gateway **20740**. Once completed, input may be received from the user selecting "analysis level" from the gateway **200** drop-down menu, to select the level of analysis in a multi-level analysis sequence to be accomplished at the local level by a gateway-resident analysis engine. The user accesses the container editor to containerize (Fig. **5**). Input is received from the user selecting the registers by clicking on the graphically depicted register, or from a drop down menu.). Input is received from the user setting the registers as described elsewhere in ("container registers"). Input is received from the user selecting or defining the rules governing the interaction of that gateway with transient containers. Input is received from the user selecting or defining the rules governing the interaction of containers existing within the logical domain of the container 100 to which that gateway is attached. The user publishes the gateway (Fig. **5**). Input is received from the user selecting "residence" from the main menu bar.). Input is received from the user choosing to leave the gateway where it was created, move it to container on another drive, directory, computer, or network. If the user chooses "move," a browse function allows the user to name the new location or browse a list of possible locations. Once input is received from the user choosing the residence of the gateway, the program restores the search interface screen.

The invention includes a data reporting means editor **610**, and a data collection means editor **710**, Fig. 2 A, as a menu option under the gateway editor **210**.

The present invention also includes a gateway process.

Referring now to FIG. **12**, a flow chart of the gateway process is shown. A system operation, search process or element container or process container is shown in transit **21100** passing through a gateway **21200**. The container, operation or process interacts with the gateway **21300**, uploading, downloading and exchanging information with the container, operation or process. The gateway stores container information **21400** and the container registers are altered **21500**. The container registers also interact with the search interface **21600**. The gateways report the register information or make it available for collection by the data reporting and collection means (Fig. **8**) operating on the communication device **26** to provide the information to the analysis engine **21800**, which stores **90100**, analyzes and instructs the execution engine **21900**, which processes and instructions are also stored **90100** by the execution engine upon receipt.

All operations in Fig. **12** utilize the communication device **26** whenever necessary or expeditious.

Referring now to FIG. **13 A,** a drawing of nested containers **100** prior to the container modification process on a network **201** is shown. (Note: The same container numbering scheme is used in Fig. 13 A, 13 B, 13 C, 13 D and in 2 B.) Information containers **505** and **909**, residing within container **908**, operating under the rules governing container interaction within that container **908** downloaded to container **505** and **909** from gateway **9081** upon their entrance to container **908,** which rules had been downloaded from execution engine **500** acting under the direction of analysis engine **400**, and under the rules programmed into their own registers **404120, 909120**, compare the specified (by those rules) set of registers **404120, 909120**, i.e., time and space, and determine a container **404** encapsulated within **505** would be more appropriately encapsulated within container **909**.

Referring now to FIG. **13 B** a drawing of nested containers during a container modification process on a network **201** is shown. Container **404** is moved to reside with container **909**. As the container **404** exits container **505**, the gateway of container **505**, being gateway **5051**, operating under the rules governing container interaction with a gateway 5051 upon egress or egress as programmed in the gateway editor **210** and modified by the execution engine **500** executing the instructions of the analysis engine **400**, or any greater logical analysis engine **408** providing execution instructions to an execution engine **508** operating in a larger encompassing container **108** entering through that container's gateway **208** or an independent gateway **707**, or sub-analysis engine operating at any gateway level, records the register information of container **404.** The gateway **5051** reports the transaction to the gateway **9081** of container **908**, being the next higher logical container. Gateway **9081** holds in gateway storage **205** the information until collected by one or more data collection processes **700**, or reported to one or more data reporting processes **600**, serving one or more analysis engines **400** residing independently on the system **10** or an analysis engine at higher logical container **303**. The analysis engine **400,** comparing reports of user hierarchically graded usage under the operations of the search engine **320** and the search interface **300**, on information container **808** after receiving reports from the data reporting means of container **404** being moved to container **909** determines, i.e., that the number of time and space relevant containers residing within container **909** is sufficient to warrant an action, and directs the execution engine **500** to copy container **909**, nested within container **908**, to a third information container **808**. As the copy instruction from execution engine **500** transits the gateway of container **908**, the gateway **9081** records the instruction. The copy instruction interacts with the registers **909120** of container **909** regarding the rules governing its copying to another location. Once approved by the governing rules of registers **909120** appended to container **909**, container **909** is duplicated. As the duplicate container **909** exits the container **908**, the gateway records the register information **909120** of container **909**, and the registers **909120** of container **909** are altered by special instructions from gateway **9081** under the rules residing in gateway **9081** regarding ingress and egress and the rules residing in the registers **909120** of container **909** regarding alteration by gateways upon ingress and egress. Passing through independent gateway **707**, the register information **909120** is recorded, and awaits data collection or reporting 700, **600.** As container **909** enters container **808**, the gateway records the register information **909120** of container **909**, the registers **909120** of **909** are altered by special instructions from gateway **8081**, operating under the rules as described in the paragraph above, and container **909** takes up residence within container **808.**

Referring now to FIG. **13 C,** a drawing of nested containers after the container modification process on a network **201** process is shown. Container **909**, now also logically residing within container **808,** commences to interact with other containers **606** in **808** under the rules governing container interaction within container **808** as received from gateway **8081** upon transiting that gateway, and under the rules of registers **606120, 909120** of the interacting containers **606, 909,** operating under the rules as described in the paragraph above. Through data collection and reporting **700, 600,** analysis engine is appraised of container's **909** new duplicate residence. I.e., operating under the registers of space relevance, a body of law pertaining to Boston Municipal tax law may be housed in a container holding Massachusetts tax law, but it would be more appropriately located in a container holding Boston tax law, with only a pointer to that location residing in the Massachusetts tax law container. In this example, such an analysis could be accomplished by comparison of zip code information in the space registers, or logical rules-based analysis, with "state" being a larger set than "city". Or, i.e., operating under the registers of time relevance, the curve of time relevance for a concert might follow an ascending curve for the months prior, hit a brief plateau, and then reach a precipitous decline, at which time certain pertinent information only might be moved to an archival container of city events or rock concerts of that year. In this example, once the curve is mapped into a register, that map would cause an increasing frequency of pointers to that container in other containers or gateways, or inclusion of that container in other containers, as the analysis engine compares that curve with increasing user inquiry.

Referring now to Fig. **13 D**, a flowchart of the reconstruction process is shown.

Information containers **505** and **909**, residing within container **908**, operating under the rules governing container interaction within that container **908** downloaded **888103** to container **505** and **909** from gateway **9081** upon their entrance to container **908**, which rules had been downloaded **888102** from execution engine **500** acting under the direction **888101** of analysis engine **400**, and under the rules programmed into their own registers **404120**, **909120**, compare **888104** the specified (by those rules) set of registers **404120**, **909120**, i.e., time and space, and determine **888105** a container **404** encapsulated within **505** would be more appropriately encapsulated within container **909**.

Container **404** is moved **888106** to reside with container **909**. As the container **404** exits container **505**, the gateway of container **505**, being gateway **5051**, operating under the rules governing container interaction with a gateway **5051** upon egress or egress as programmed in the gateway editor **210** and modified **888108** by the execution engine **500** executing the instructions of the analysis engine **400**, or any greater logical analysis engine **408** providing execution instructions **888107** to an execution engine **508** operating in a larger encompassing container **108** entering through that container's gateway **208** or an independent gateway **707**, or sub-analysis engine operating at any gateway level, records **888109** the register information of container **404**, and alters the register information of container **404**. The gateway **5051** reports **888110** the transaction to the gateway **9081** of container **908**, being the next higher logical container. Gateway **9081** holds **888111** in gateway storage **205** the information until collected by one or more data collection processes **700**, or reported to one or more data reporting processes **600**, serving **888112** one or more analysis engines **400** residing independently on the system **10** or an analysis engine at higher logical container **303**. The analysis engine **400**, comparing **888114** reports of user hierarchically graded usage on information container **808** under the operations of the search engine **320** and the search interface **300**, after receiving **888113** reports from the data reporting means of container **404** being moved to container **909**, determines **888115**, i.e., that the number of time and space relevant containers residing within container **909** is sufficient to warrant an action, and directs **888115** the execution engine **500** to copy container **909**, nested within container **908**, to a third information container **808**. As the copy instruction from execution engine **500** transits the gateway of container **908**, the gateway **9081** records **888116** the instruction. The copy instruction interacts **888117** with the registers **909120** of container **909** regarding the rules governing its copying to another location. Once approved **888118** by the governing rules of registers **909120** appended to container **909**, container **909** is duplicated **888118**. As the duplicate container **909** exits the container **908**, the gateway records **888119** the register information **909120** of container **909**, and the registers **909120** of container **909** are altered **888120** by special instructions from gateway **9081** under the rules residing in gateway **9081** regarding ingress and egress and the rules residing in the registers **909120** of container **909** regarding alteration by gateways upon ingress and egress. Passing through independent gateway **707**, the register information **909120** is recorded **888121**, and awaits **888122** data collection or reporting **700, 600.** As container **909** enters container **808**, the gateway records **888123** the register information **909120** of container **909**, the registers **909120** of **909** are altered **888124** by special instructions from gateway **8081**, operating under the rules as described in the paragraph above, and container **909** takes up residence **888125** within container **808.**

Container **909**, now also logically residing (in addition to its original container residence) within container **808,** commences to interact **888126** with other containers **606** in **808** under the rules governing container interaction within container **808** as received from gateway **8081** upon transiting that gateway, and under the rules of registers **606120, 909120 of** the interacting containers **606, 909,** operating under the rules as described in the paragraph above. Through data collection and reporting **700, 600,** analysis engine is appraised **888127** of container's **909** new duplicate residence.

Referring now to Fig. **14**, the screen interface of the container editor is shown. This interface is a process wherein input is received by the user using the main menu **78** or drop down menu **1419**, or using an input device to "drag and drop" or click, causing the system **10** to acquire **1409**, edit **1410** or create **1411** a file **1407**, container **1408** or digital content **01**, to search for **1412**, acquire **1413**, edit **1414** or create **1415**, print **1416**, or containerize **1417** a container **100**, to select **1402**, (or by clicking on register), search **1403**, acquire **1404**, edit **1405**, or create a register **1406** to append or detach registers **120** to those containers, to set register values in those registers **120**, to utilize the register editor **125** through **1405** to create new registers, or to **1418** add, detach, acquire a gateway **200** to append or detach to those containers, and utilize the gateway editor **210** through **1418**. (See detailed description referring to Fig. 5)

Referring now to Fig. **15**, the screen interface of the gateway editor is shown. This interface is a process wherein input is received by the user using the main menu **1501** or drop down menu **1513**, or using an input device to "drag and drop" or click, causing the system **10** to search for **1507**, acquire **1508**, edit **1509** create **1510**, print **1511** or containerize **1512** gateways, and causing the system **10** to establish rules by which an individual gateway governs the transiting **1502**, entering **1503**, exiting **1504** of containers and the interaction of containers within its domain **1505**, and external of its domain.**1506**. (See detailed description referring to Fig. **11**).

Referring now to Fig. **16**, the screen interface of the search interface. This interface is a process wherein input is received by the user using the main menu **1625** or drop down menu **1624**, or using an input device to "drag and drop" or click, or by entering text, causing the system **10** to select **1615**, search for **1616**, acquire **1617**, edit **1618** create **1619**, print **1620**, containerize **1621** (by accessing the container editor **110**) or insert **1622** digital search strings into the search box **1623** in order to submit that string to the search engine **320**, or causing the system **10** to select **1602**, search for **1603**, acquire **1604**, edit **1605**, create **1612**, containerize **1613** (by accessing the container editor **110**), or insert **1614** search keys (templates that comprise search scope in geographic range, container level, and specific key words or digital strings), or containerized searches (containers **110**), into the search box **1623** in order to submit that string to the search engine **320** , or causing the system **10** to set a search range by geographic range **1607**, container level **1608**, or acquire **1609**, edit **1610** or create **1611** a scope template. (templates that comprise search scope in geographic range and, container level.) (See detailed description referring to Fig. 6).

Referring now to Fig. **17**, a drawing showing, on an input device or computer screen **24**, in any generic (dashed lines) software application program, a drop-down menu link **1403** on a drop down menu **1402** dropping down from a main menu **1401**, and a free-floating button link **1404**, is shown. When input is received at **1402** or **1403**, the system **10** makes available to the user the containerization process or container editor **110**. When input is received at drop-down menu link **1405** or a button link **1406**, the system **10** makes available to the user the means to enter and interact with this system **10** or this network **201** in any of their aspects. The interfaces **1403, 1404** show a process wherein input is received causing the system **10** to encapsulate content or access the container editor **110**. The link also allows the user to encapsulate the page or file on which he is currently working, without selecting content, and if so desired, without accessing the container editor. The interfaces **1405, 1406** show a process wherein input is received causing the system **10** to access or interact with the system **10** or the network **201**.

The present invention also includes a search engine **320**. Once the key word(s), phrase or digital segment is entered into the search interface **300**, or an offered selection chosen on the menu, it is utilized by the search engine **320** to locate the desired site or data.

The search engine employed may be any industry standard search engine such as Verity "Topic", or Personal Library Software, as used in Dow Jones News Retrieval, or Internet search engines such as Webcrawler, Yahoo, Excite, Infoseek, Alexa or any Internet search engine, or any new engines to be developed capable of searching for and locating digital segments, whether text, audio, video or graphic.

The present invention also includes an analysis engine **400**. Utilizing rules-based analysis, the analysis engine determines the class of storage medium upon which containers reside, the subsets and supersets by which and in which containers encompass and reside within one another, the routes of access to those containers, the historically successful search parameters by which those containers are accessed based upon the identity of the user accessing the containers, and the grade of access chosen by the user in accessing that container **100**.

Utilizing a pre-programmed sequence of compilation, and inductive, deductive and derivative analysis, the analysis engine manufactures instructions based upon the analysis of the information submitted by the gateways and the search interface, and submits those instructions to the appropriate execution engine **500** in order to create new information containers, content assemblages, storage schemes, access routes, search templates, and gateway instructions, and others, and to provide informed search options through the search interface to the inquiring user.

The present invention also includes an engine editor **510**, that provides a system administrator with a means of editing the operating principles of that search engine, and search template loading in the search interface **300**, a reporting and collection means editor **610**, **710**, governing data reporting **600** and data collection **700** at the gateways **200** as defined by the gateway editor **210** and the register editor **125**, a container editor **110** for creating and modifying containers and appending registers to containers, a register editor **125** for creating and modifying container registers and establishing and adjusting the values therein, container gateways **200** with their own storage **205**, information containers **100** for holding information and container registers for holding information about specific containers and their history on the network.

The present invention also includes an execution engine **300**. Based upon instructions received from the analysis engine **400** utilizing the communication device **26**, the execution engine **500** provides search phrases to the search interface **300** based upon initially received inquiries, relocates containers including their programs, data and registers to other directories, drives, computers, networks on other classes of storage mediums, i.e., tape drive, optical drive, CD-ROM, deletes, copies, moves containers to nest within or encompass other containers on other directories, drives, computers, networks to nest within other containers, alters the class of storage medium upon which containers reside, the subsets and supersets by which and in which containers encompass and reside within one another, the routes of access to those containers, and the historically successful search parameters by which those containers are accessed based upon the identity of the user accessing the container and the grade of access chosen by the user in accessing that container.

The execution engine **400** fulfills the instructions of the analysis search engine **500**, to create new information containers, content sub and superset assemblages, storage schemes, access routes, search templates, gateway **200** instructions and other system functions. The execution engine includes an editor **510** that provides a system manager with a means of editing the operating principles of that search engine, governing data reporting, data collection **700**, search template loading, gateway instructions, and other functions.

The present invention also includes flat or relational databases **900**, used where, and as required.

The present invention also includes a communication device **26** supporting all operations on a network wide basis.

The present invention also includes a search engine **300** to locate the desired site or data. The present invention also includes databases **900**, flat or relational, to serve the other components of the system as needed and where needed.

The present invention also includes editors, by which the user may alter the governing aspects of the system. Editors include, but are not limited to, a container editor **110**, a register editor **125**, a gateway editor **210**, an engine editor **510**, a reporting means editor **610**, a search interface **300**, and a collection means editor **710**.

The present invention also includes specific screen interfaces for the editors, as described in Fig. **14**, Fig. **15**. and Fig. **16**.

The present invention also includes a means for this system **10** and network **201 or container editor 110** to be accessed from a menu or button selection within any program, as described in Fig. **17**.

While the present invention has been described with reference to certain preferred embodiments, those skilled in the art will recognize that various modifications may be provided. For example, both analysis engine and execution engine may be duplicated or modified for distribution at various locations and hierarchical positions in the gateway and container system throughout the network and designed to work in concert. Also, the physical computing infrastructure may be mainframe, mini, client server or other with various network and distributed computing designs, including digitally supported or based physical or public media, and the components of the system 10, as described in Fig. 1 may be physically distributed through space. Even the contents of a single container may be logically referenced but be physically distributed through the network and reside at multiple storage locations. The whole system may be hierarchically nested within other systems to the nth degree. Whole systems may also be encapsulated within containers. A single container may also encompass a single physical media, such as a CD-ROM disk, programmed with the container, gateway and register design. Gateways may be strategically placed on containers at ingress and/or egress points or may be placed strategically throughout the system for optimal collection and reporting output and gateway system control. Also, the loop of gateway data collection and reporting, analysis engine analysis, instruction, and gateway modification, and execution engine operations may be infinitely nested, from the smallest container of two sub-containers to whole networks holding millions of containers and thousands of levels, with analysis itself nested within the multiple levels. Gateways may be established at both logical and physical junctures such as a satellite uplink point. Also, the provision to establish a unique network identity might be designed to include as of yet unknown computer networks as they arise. The analysis and execution engines may operate on a rules-based, fuzzy logic, artificial intelligence, neural net, or other system not yet devised. Other variations upon and modifications to the preferred embodiments are provided for by the present invention, which is limited only by the following claims. Also, the classification scheme of nested containers, while designated by the container creators, may transform, be utilized otherwise, or be wholly discarded according to usage. Also, hardware configurations, such as the use of RAM or hard drives for storage or lasers for communication may assume myriad forms without altering the essential operation of this invention.

WHAT IS CLAIMED IS:

1. An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element;

a plurality of registers, the plurality of register forming part of the container, a first register of the plurality of registers for storing a unique container identification value, a second register of the plurality of registers that stores information and evolves according to the relationship, use and interaction of the container with other containers, processes and systems; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems and process.

2. The apparatus of claim 1, wherein the information element is one from the group of text, graphic images, video, audio, a digital pattern, a process, a nested container, bit, natural number and a system.

3. The system of claim 1, wherein the plurality of registers include at least one container history register for storing information regarding past interaction of the container with other container, system or processes, the container history register being modified.

4. The system of claim 1, wherein the plurality of registers include at least one system history register for storing information regarding past interaction of the container with different operating system and network processes.

5. The system of claim 1, wherein the plurality of registers include at least one predefined register, the predefined register being a register associated with an editor for user selection, the predefined register appendable to any container.

6. The system of claim 1, wherein the plurality of registers include a user-created register, the user-created register generated by the user, one or more of which is appendable to any container.

7. The system of claim 1, wherein the plurality of registers include a system-defined register, the system-defined register set, controlled and used by the system, one or more of which is appendable to any container.

8. The system of claim 1, wherein the plurality of registers include at least one register for controlling the relationship of the container with other containers, systems and processes using time as a parameter.

9. The system of claim 8, wherein the plurality of registers include:

an active time register for identifying times at which the container will act upon other containers, processes, systems or gateways;

an passive time register for identifying times at which the container can be acted upon other containers, processes systems, or gateways; and

a neutral time register for identifying times at which the container may interact with other containers.

10. The system of claim 1, wherein the plurality of registers include at least one acquire register for controlling whether the container adds a register or a container from other containers when interacting with them.

11. The system of claim 1, wherein the plurality of registers include at least one register for controlling the relationship of the container with other containers using space as a parameter.

12. The system of claim 11, wherein space refers to the geographic location of a the container.

13. The system of claim 11, wherein space refers to the logical address space of a network in which a container resides.

14. The system of claim 11, wherein the plurality of registers include:

an active space register for identifying space in which the container will act upon other containers, processes, systems or gateways;

an passive space register for identifying from which the container can be acted upon other containers, processes systems, or gateways; and

a neutral time register for identifying space in which the container may interact with other containers.

15. The system of claim 1, wherein the gateway includes means for acting upon another container, the means for acting upon another container using the plurality of register to determine whether and how the container acts upon other containers.

16. The system of claim 1, wherein the gateway includes means for allowing interaction, the means for allowing interaction using the plurality of registers to determine whether and how another container can act upon the container.

17. The system of claim 1, wherein the gateway includes means for gathering information, the means for gathering information recording register information from other containers, systems and processes that interact with the container.

18. The system of claim 1, wherein the gateway includes means for reporting information, the means for reporting information providing register information to other containers, systems and processes that interact with the container.

19. The system of claim 1, wherein the gateway includes an expert system including rules defining the interaction of the container with other containers, systems and processes.

20. A method for creating an interactive information container, the method including the steps of:

forming a container;

selecting an interactive register for the container;

identifying an item for inclusion in a container; and

creating a container element that includes the identified item.

21. The method of claim 20, wherein the step of forming a container further comprising the steps of:

displaying a plurality of container levels;

receiving input from a user selecting one of the displayed container level; and

displaying a container template corresponding to the container level input.

22. The method of claim 20, wherein the step of selecting an interactive register further comprising the steps of:

displaying a list of available registers;

receiving input selecting an available register from the list of available registers;

receiving input values for the selected available register; and

appending the register to the container.

23. The method of claim 20, wherein the step of creating a container element that includes the identified item further comprising the steps of:

providing a data structure as part of the container element;

storing the identified element in the data structure; and

associating the container element with the container.

24. The method of claim 20, wherein the step of forming a container includes the step of providing for the container a gateway that uses the interactive register to control the interaction of the container with other containers, processes, and systems.

25. The method of claim 24, wherein the step of providing a gateway further comprising the steps of:

determining a current gateway for a system upon which the container is being created;

replicating the current gateway to create a new gateway; and

associating the new gateway with the container.

26. The method of claim 24, wherein the step of providing a gateway further comprising the steps of:

determining a register for a system upon which the container is being created;

replicating the determined register to create a new register; and

associating the new register with the container.

27. The method of claim 20, wherein the step of selecting an interactive register further comprising the steps of:

retrieving a list of available registers;

selecting an available register from the list of available registers by the system;

receiving input values for the selected available register from the system; and

appending the register to the container.

28. The method of claim 20, wherein the step of creating a container element that includes the identified item is performed by a system interacting with the container, and further comprising the steps of:

providing a data structure as part of the container element;

storing the identified element in the data structure; and

associating the container element with the container.

29. A method for interacting between a first interactive information container and a second interactive information container, the method including the steps of:

determining identification information for the first container using a first gateway;

determining identification information for the second container using a second gateway;

determining whether the first container can act upon the second container using the first gateway and a register of the first container;

determining whether the second container can be acted upon by the first container using the second gateway and a register of the second container; and

performing the interaction between the first and second containers prescribed by the first gateway and the register of the first container if both the first container can act upon the second container and the second container can be acted upon by the first container.

30. The method for interacting of claim 29, wherein the steps of determining identification information are performed by reading respective identification registers of the first and second containers.

31. The method for interacting of claim 29, further comprising the step of altering a register of the first container and a register of the second container to reflect the interaction between the first container and the second container.

32. The method for interacting of claim 29, further comprising the step of adding registers to the first container based on the registers in the second container and the second gateway.

33. The method for interacting of claim 29, wherein the step of performing also uses the second gateway and the register of the second container to determine the prescribe action to be taken.

34. The method for interacting of claim 29, further comprising the steps of:

determining whether the first container should add an identified register of the second container as a new register of the first container using an acquire register and the first gateway of the first container; and

adding the new register to the first container if it is determined that the new register should be added to the first container.

35. The method for interacting of claim 29, further comprising the step of modifying the first gateway of the first container based on the interaction between the first container and the second container.

36. The method for interacting of claim 35, wherein the step of modifying includes modifying rules of an expert system that forms the first gateway of the first container.

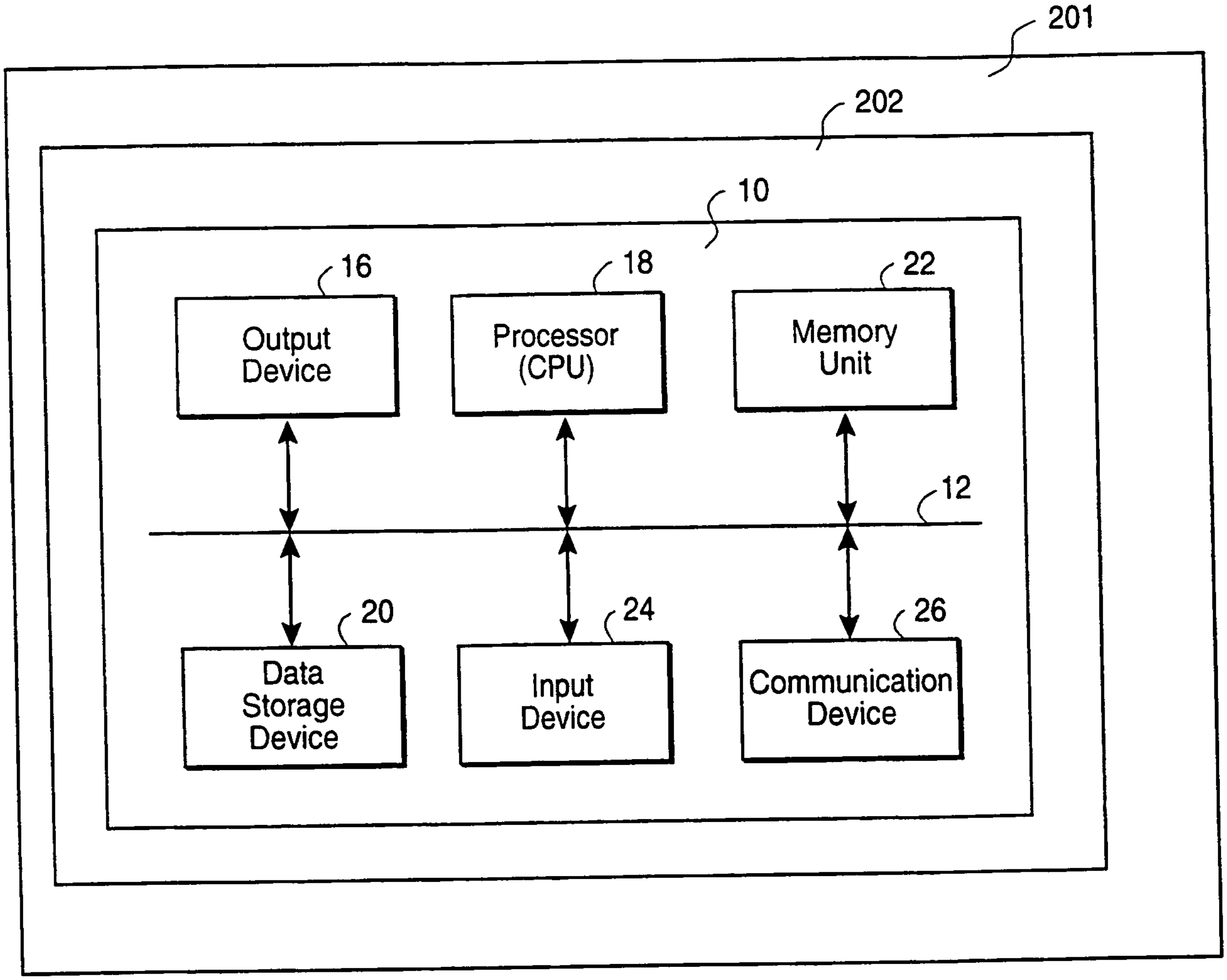

201
202
10
16
Output Device
18
Processor (CPU)
22
Memory Unit
12
20
Data Storage Device
24
Input Device
26
Communication Device

FIG. 1

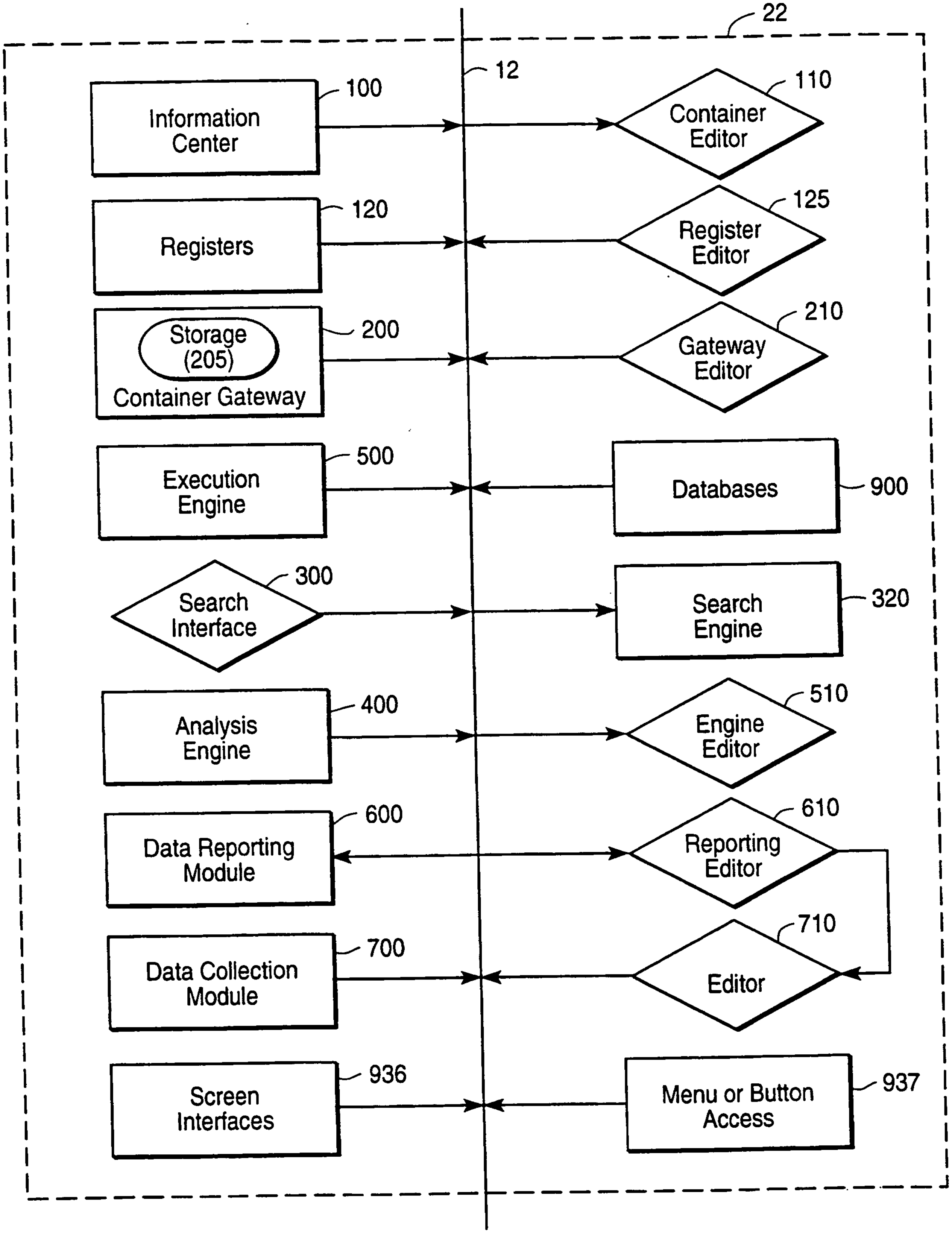

22
12
Information Center
100
Container Editor
110
Registers
120
Register Editor
125
Storage (205)
Container Gateway
200
Gateway Editor
210
Execution Engine
500
Databases
900
Search Interface
300
Search Engine
320
Analysis Engine
400
Engine Editor
510
Data Reporting Module
600
Reporting Editor
610
Data Collection Module
700
Editor
710
Screen Interfaces
936
Menu or Button Access
937

FIG. 2A

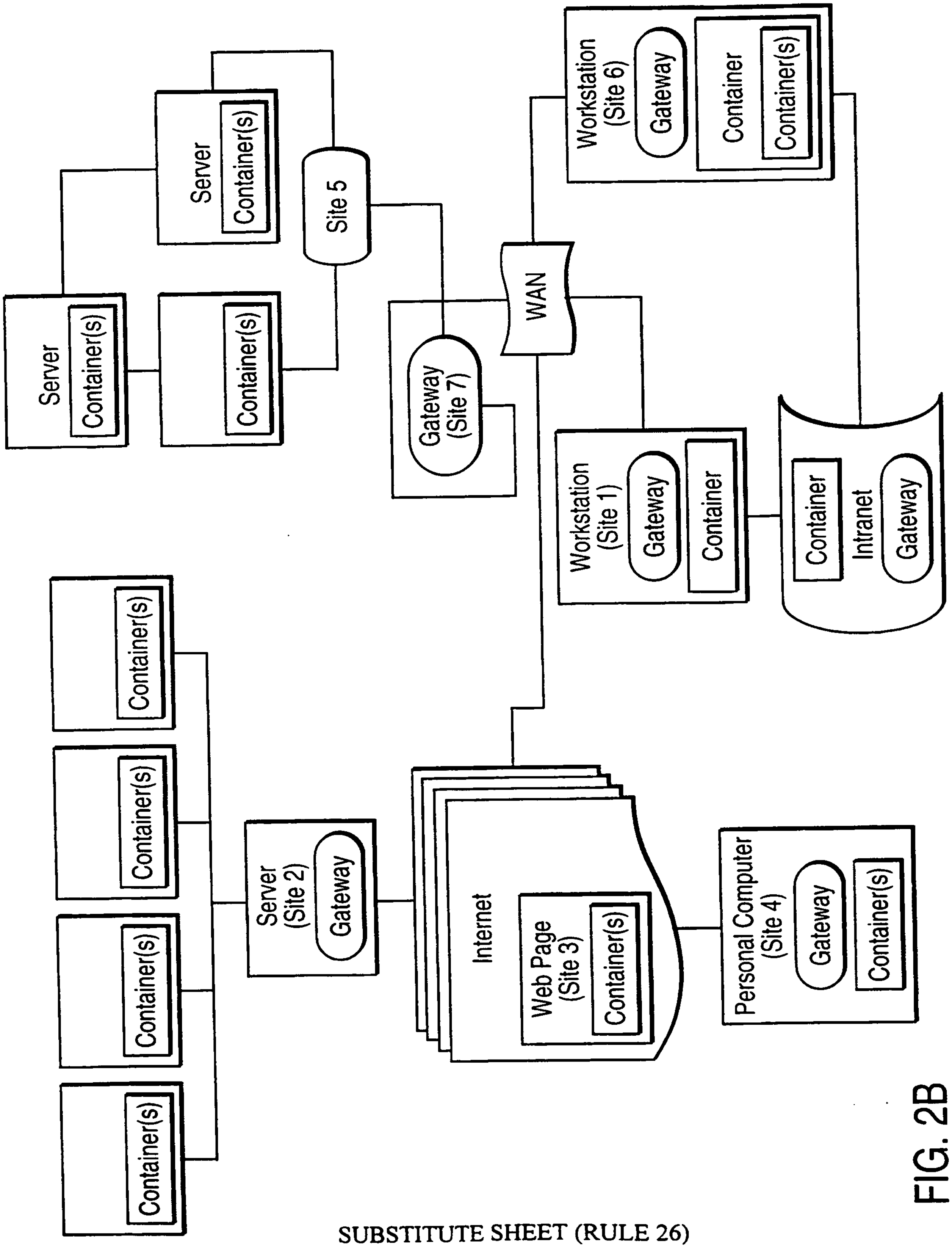

Server
Container(s)
Container(s)
Server
Container(s)
Site 5
Gateway (Site 7)
WAN
Workstation (Site 6)
Gateway
Container
Container(s)
Workstation (Site 1)
Gateway
Container
Container
Intranet
Gateway
Container(s)
Container(s)
Container(s)
Container(s)
Server (Site 2)
Gateway
Internet
Web Page (Site 3)
Container(s)
Personal Computer (Site 4)
Gateway
Container(s)

FIG. 2B

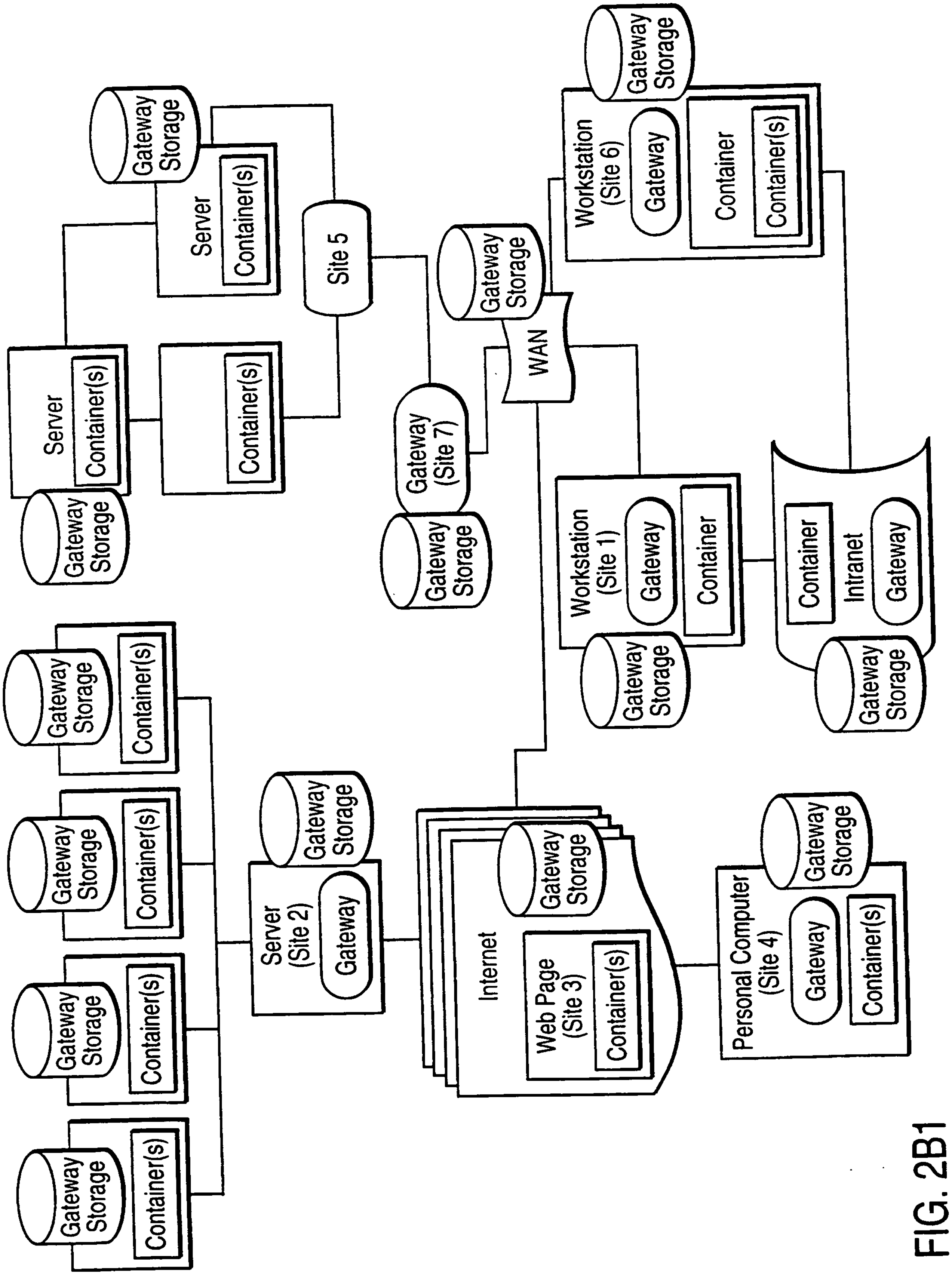

Gateway Storage
Container(s)
Gateway Storage
Container(s)
Gateway Storage
Container(s)
Gateway Storage
Container(s)
Server (Site 2)
Gateway
Gateway Storage
Server
Container(s)
Gateway Storage
Container(s)
Server
Container(s)
Gateway Storage
Site 5
Gateway (Site 7)
Gateway Storage
Internet
Web Page (Site 3)
Container(s)
Gateway Storage
WAN
Gateway Storage
Workstation (Site 1)
Gateway
Container
Gateway Storage
Workstation (Site 6)
Gateway
Container
Container(s)
Gateway Storage
Intranet
Container
Gateway
Gateway Storage
Personal Computer (Site 4)
Gateway
Container(s)
Gateway Storage

FIG. 2B1

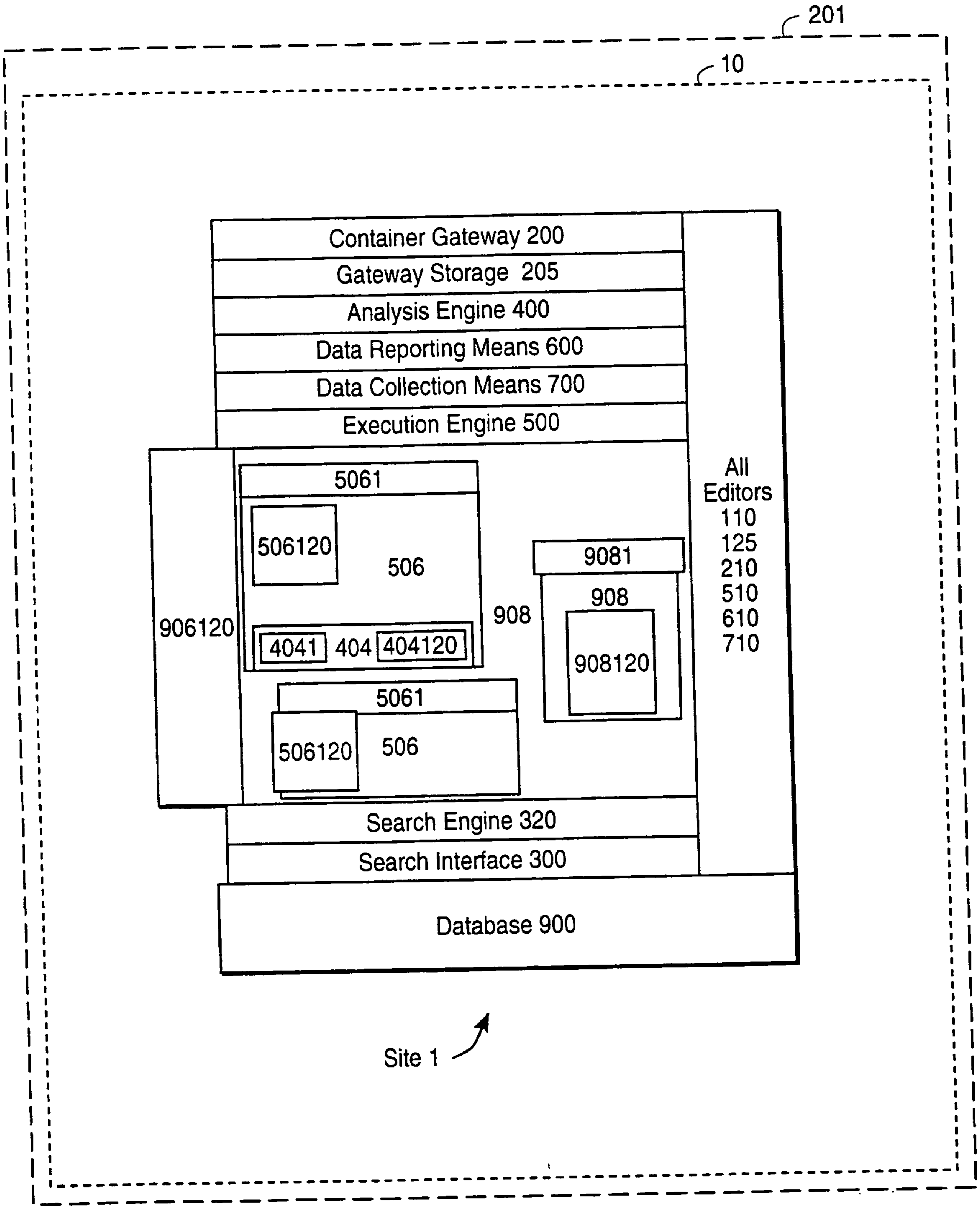

201
10
Container Gateway 200
Gateway Storage 205
Analysis Engine 400
Data Reporting Means 600
Data Collection Means 700
Execution Engine 500
906120
5061
506120
506
4041
404
404120
908
9081
908
908120
5061
506120
506
All Editors
110
125
210
510
610
710
Search Engine 320
Search Interface 300
Database 900
Site 1

FIG. 2C

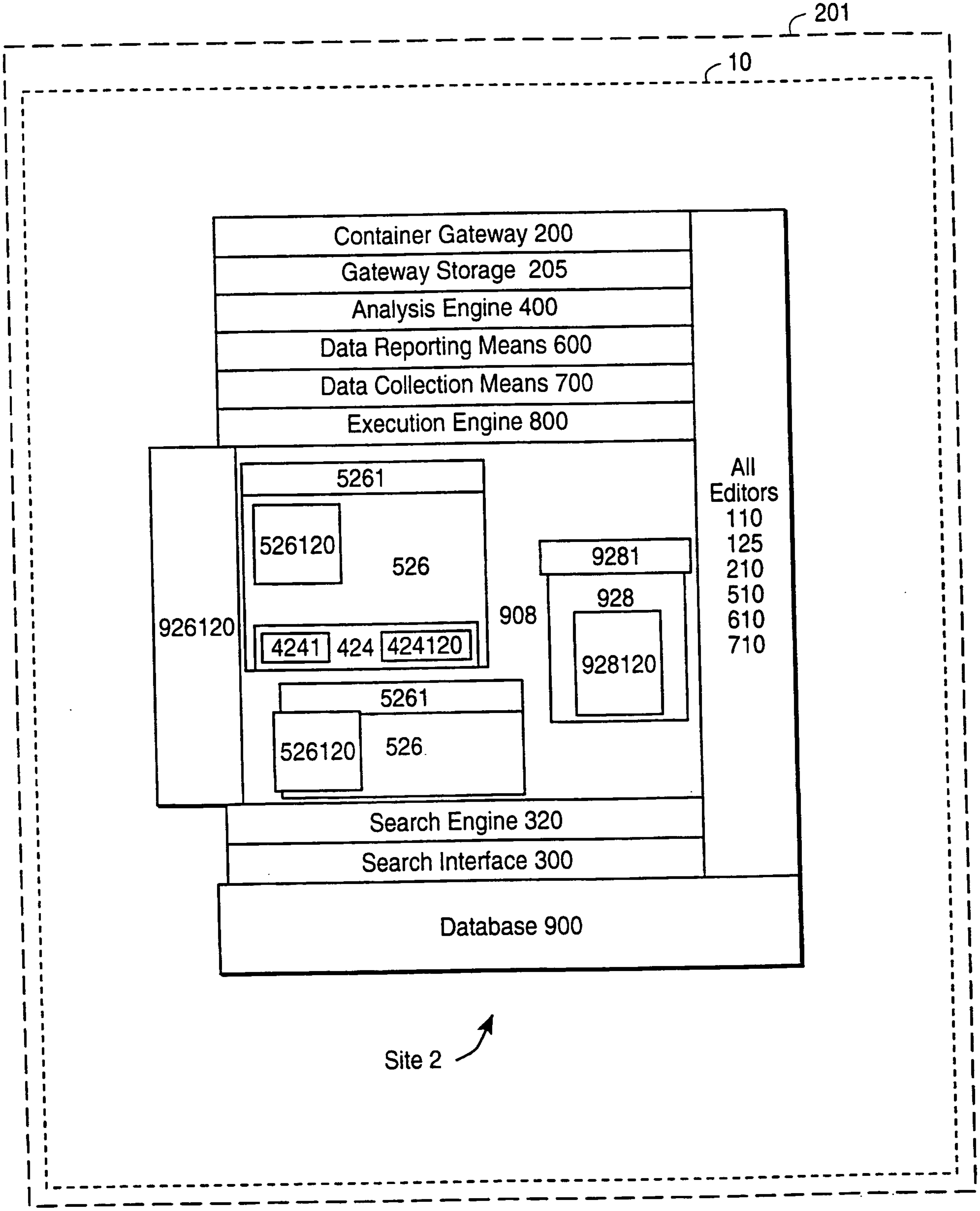

201
10
Container Gateway 200
Gateway Storage 205
Analysis Engine 400
Data Reporting Means 600
Data Collection Means 700
Execution Engine 800
926120
5261
526120
526
4241
424
424120
5261
526120
526
908
9281
928
928120
All Editors 110 125 210 510 610 710
Search Engine 320
Search Interface 300
Database 900
Site 2

FIG. 2D

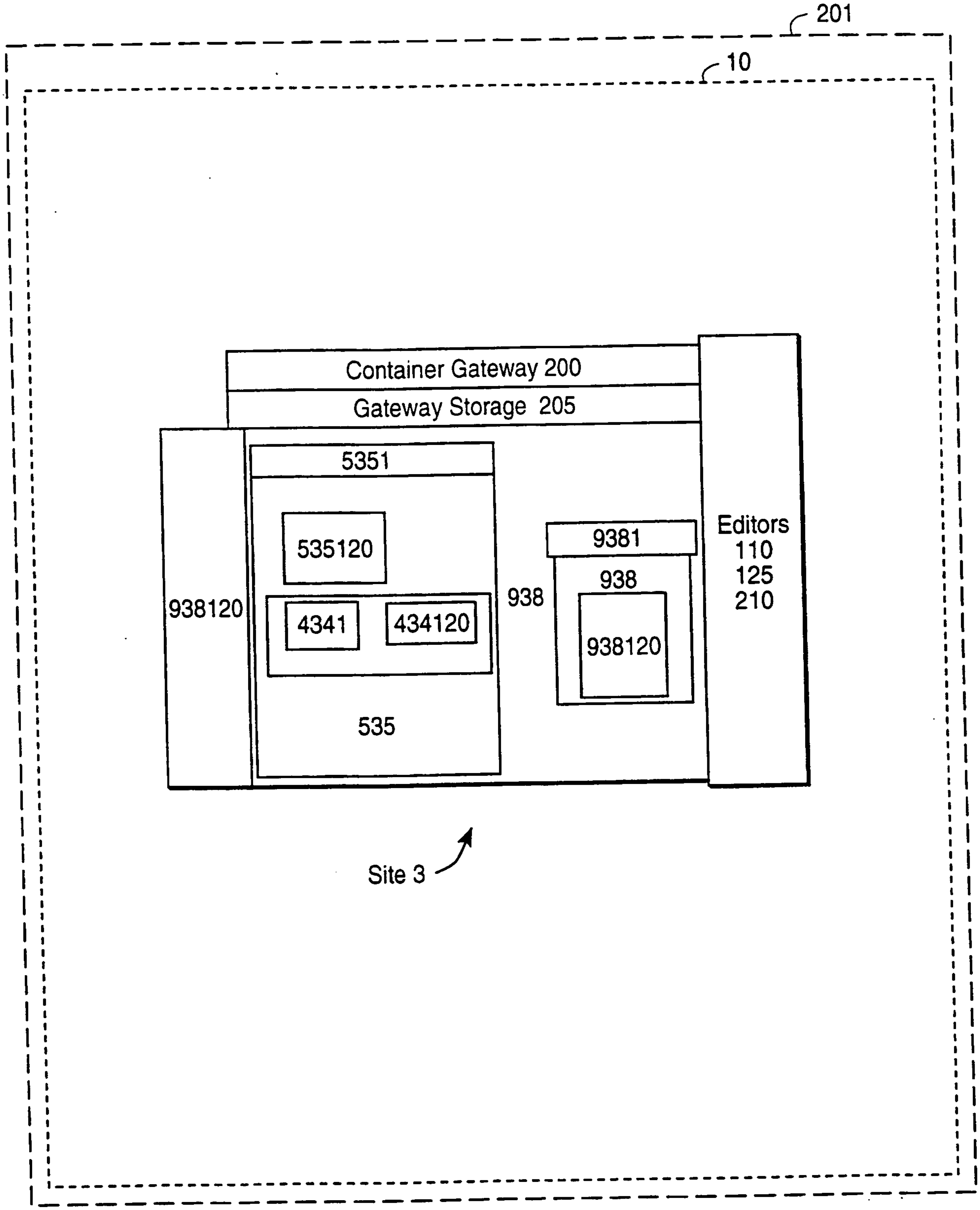

201
10
Container Gateway 200
Gateway Storage 205
938120
5351
535120
4341
434120
535
938
9381
938
938120
Editors
110
125
210
Site 3

FIG. 2E

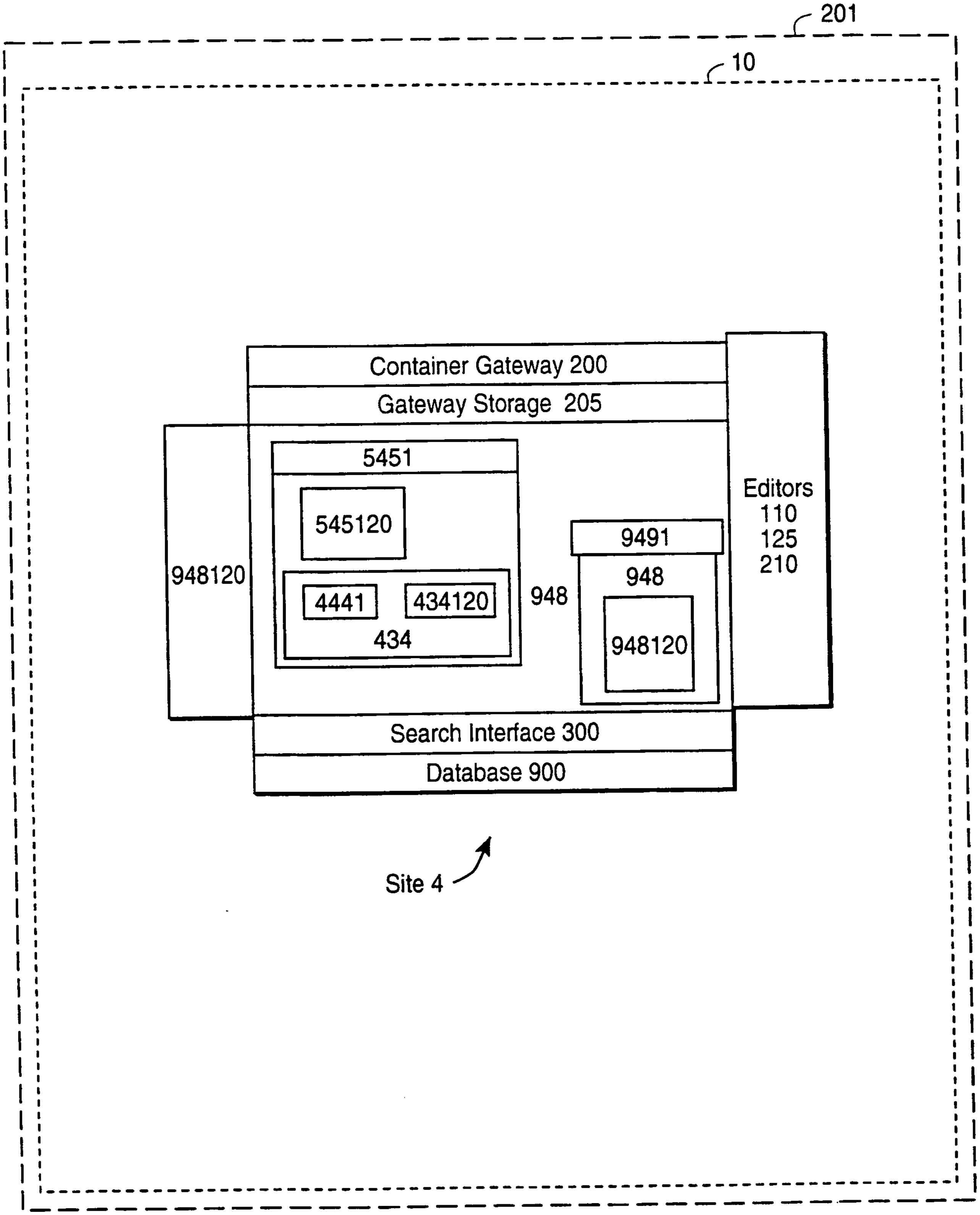

201
10
Container Gateway 200
Gateway Storage 205
5451
545120
948120
4441
434120
434
948
9491
948
948120
Editors
110
125
210
Search Interface 300
Database 900
Site 4

FIG. 2F

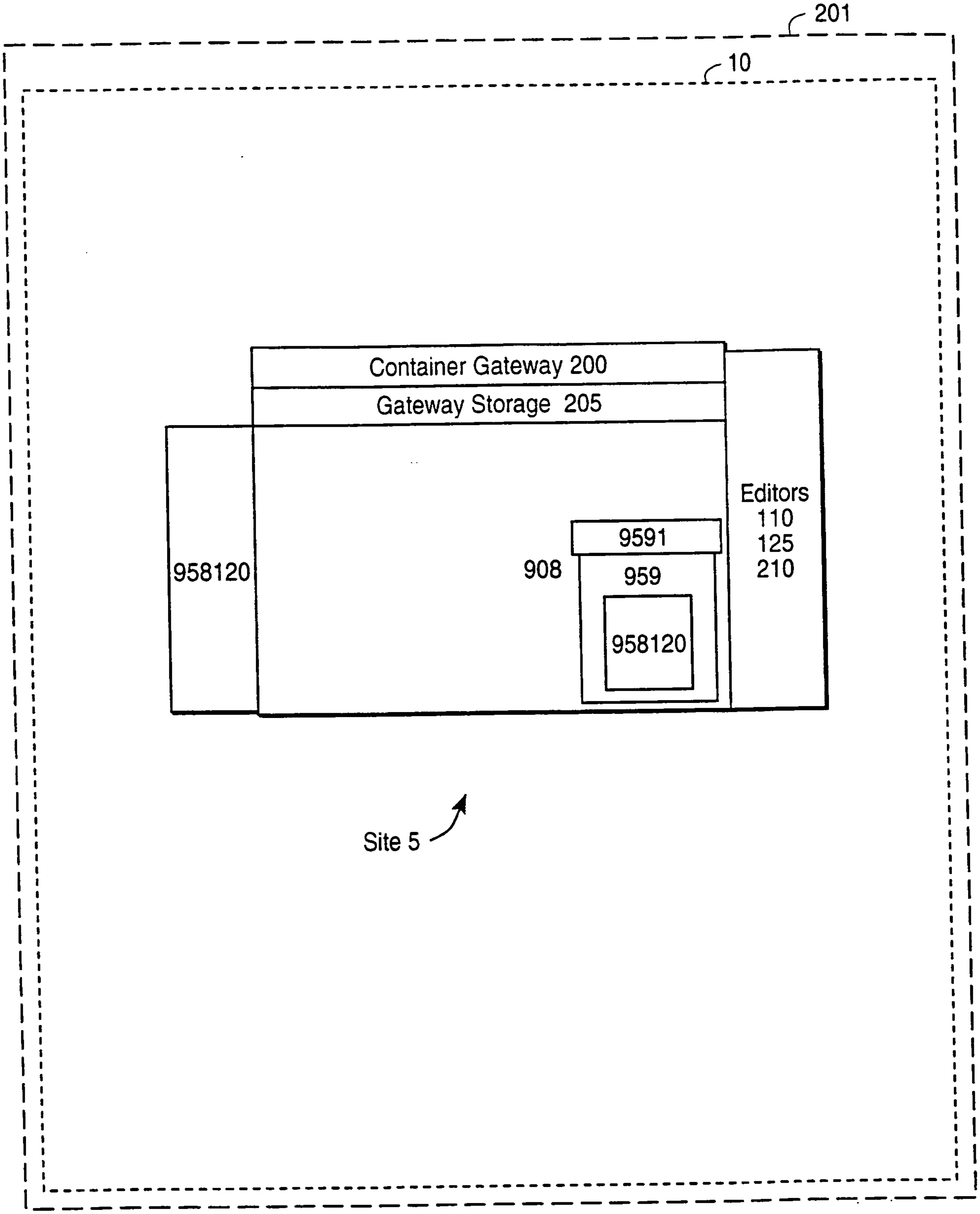

201
10
Container Gateway 200
Gateway Storage 205
958120
908
9591
959
958120
Editors
110
125
210
Site 5

FIG. 2G

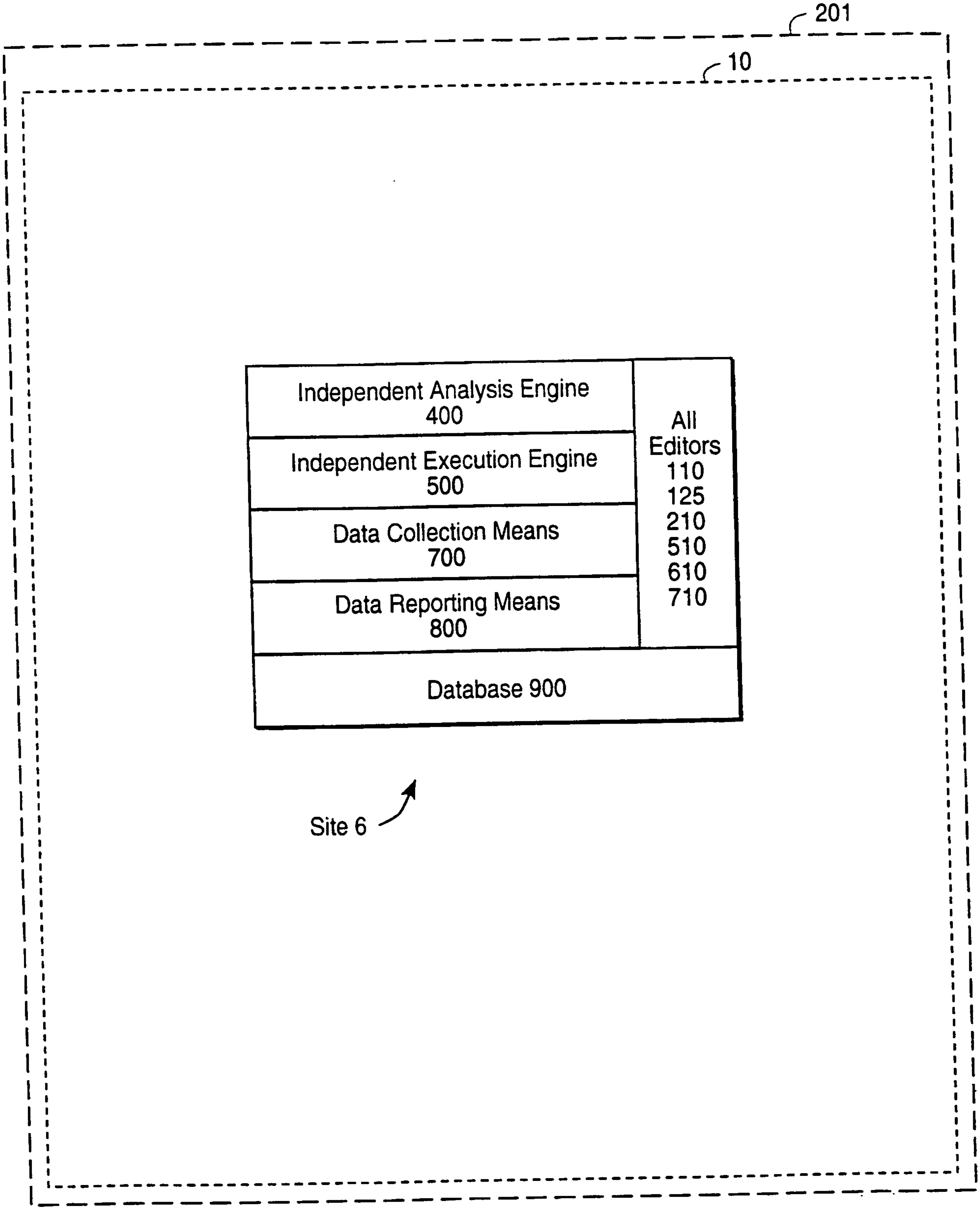

201
10
Independent Analysis Engine
400
Independent Execution Engine
500
Data Collection Means
700
Data Reporting Means
800
All
Editors
110
125
210
510
610
710
Database 900
Site 6

FIG. 2H

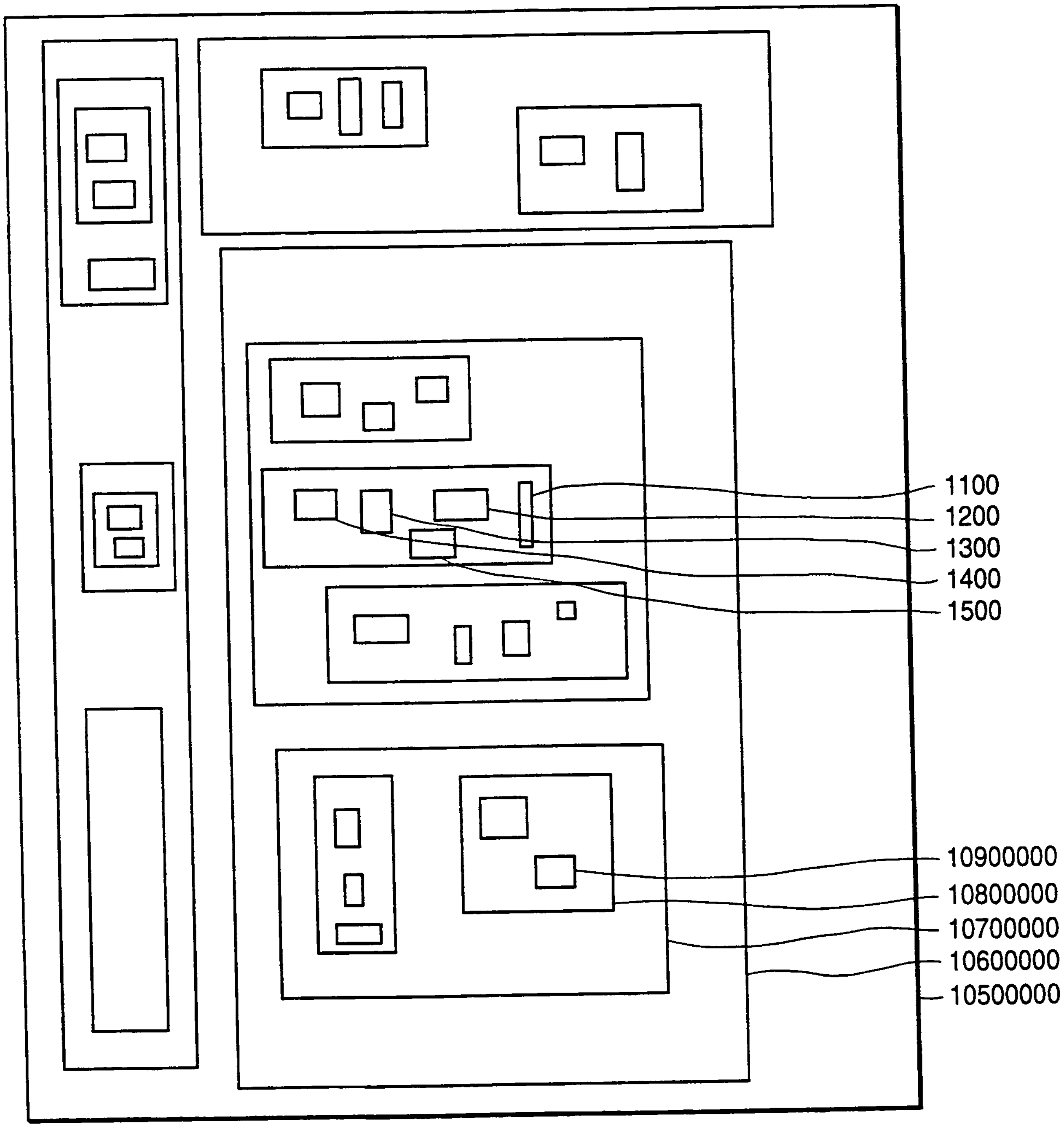

1100
1200
1300
1400
1500
10900000
10800000
10700000
10600000
10500000

FIG. 3A

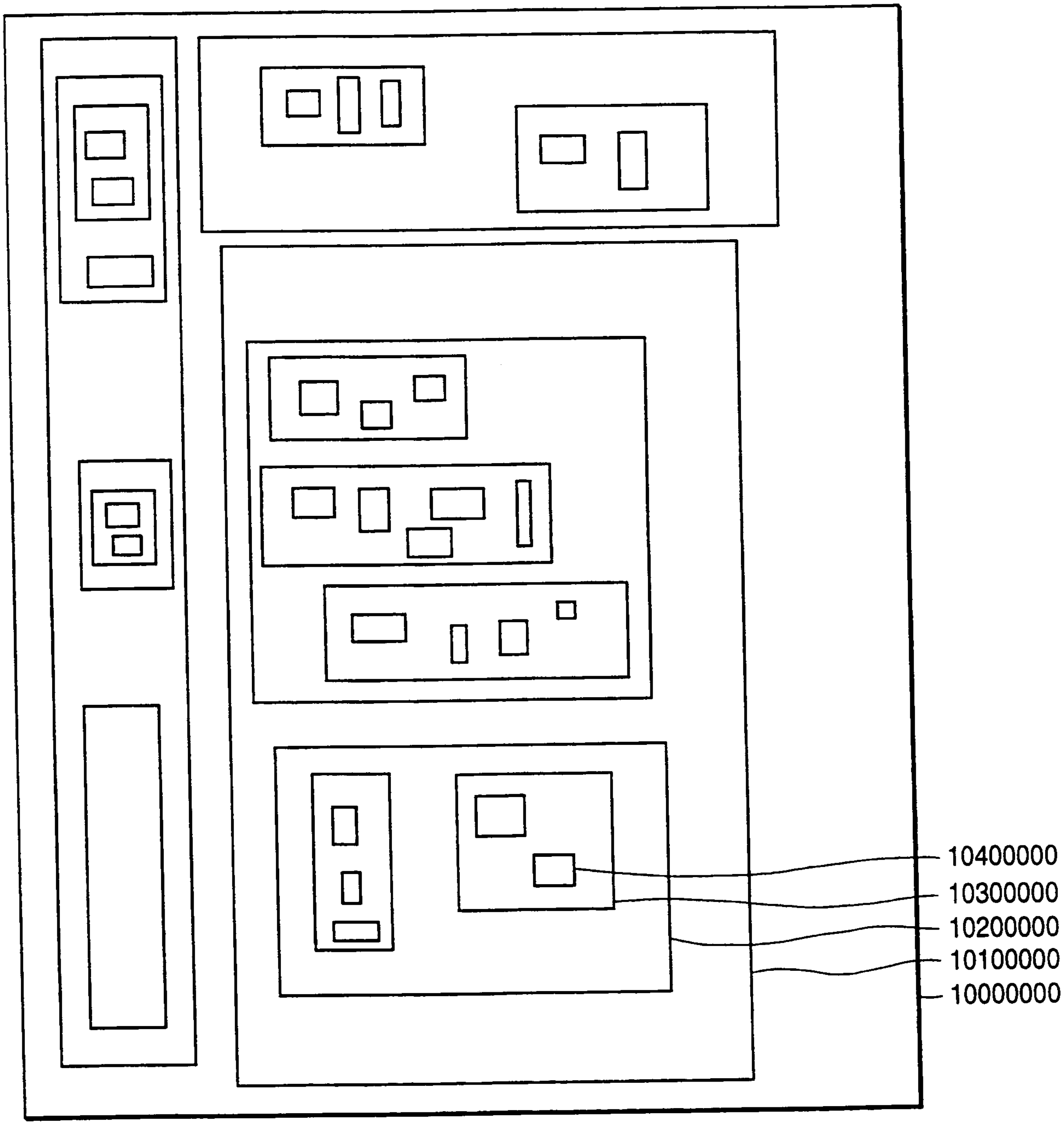


FIG. 3B

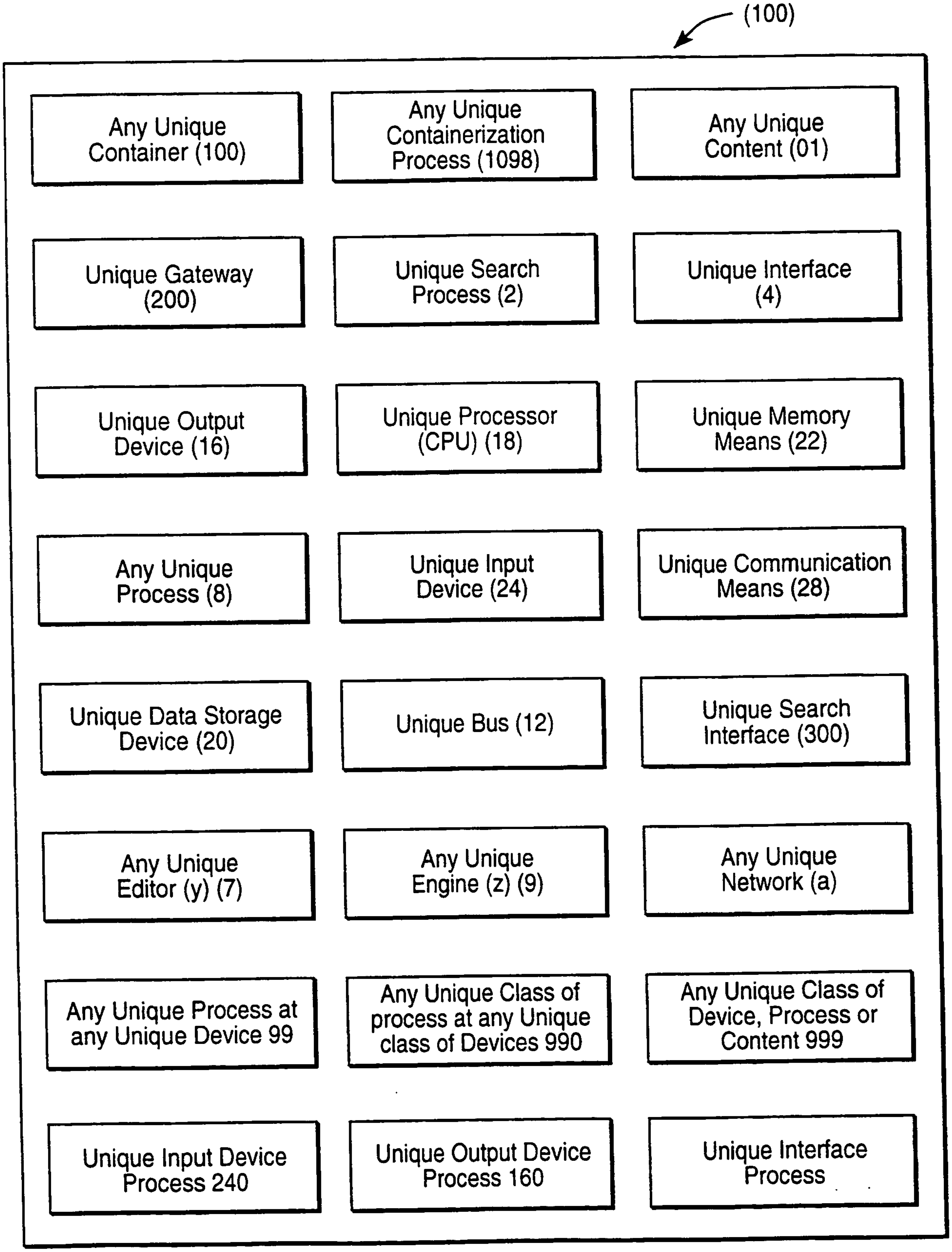

(100)
Any Unique Container (100)
Any Unique Containerization Process (1098)
Any Unique Content (01)
Unique Gateway (200)
Unique Search Process (2)
Unique Interface (4)
Unique Output Device (16)
Unique Processor (CPU) (18)
Unique Memory Means (22)
Any Unique Process (8)
Unique Input Device (24)
Unique Communication Means (28)
Unique Data Storage Device (20)
Unique Bus (12)
Unique Search Interface (300)
Any Unique Editor (y) (7)
Any Unique Engine (z) (9)
Any Unique Network (a)
Any Unique Process at any Unique Device 99
Any Unique Class of process at any Unique class of Devices 990
Any Unique Class of Device, Process or Content 999
Unique Input Device Process 240
Unique Output Device Process 160
Unique Interface Process

FIG. 3C

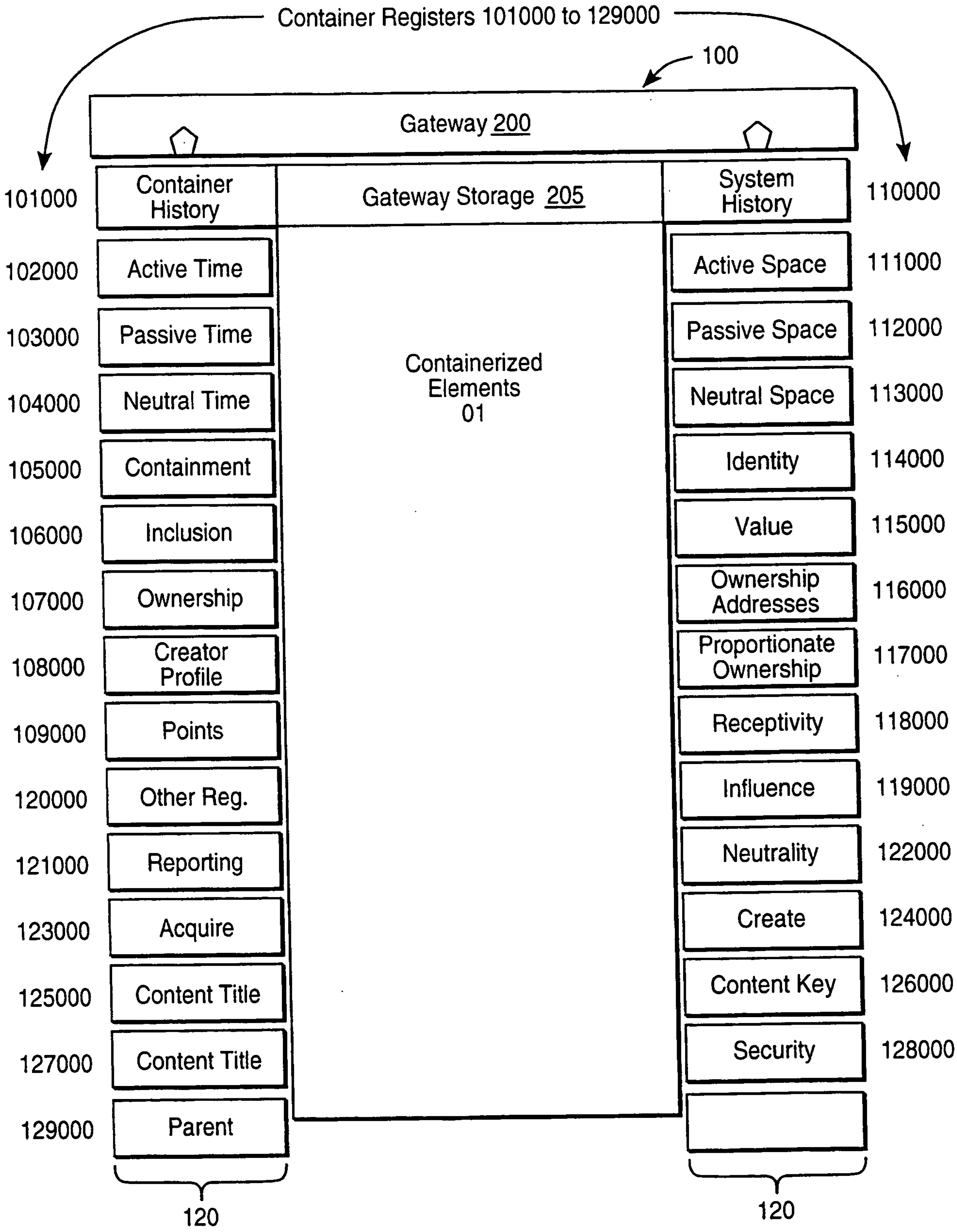

Container Registers 101000 to 129000
100
Gateway 200
Gateway Storage 205
Containerized Elements 01
101000
Container History
102000
Active Time
103000
Passive Time
104000
Neutral Time
105000
Containment
106000
Inclusion
107000
Ownership
108000
Creator Profile
109000
Points
120000
Other Reg.
121000
Reporting
123000
Acquire
125000
Content Title
127000
Content Title
129000
Parent
120
System History
110000
Active Space
111000
Passive Space
112000
Neutral Space
113000
Identity
114000
Value
115000
Ownership Addresses
116000
Proportionate Ownership
117000
Receptivity
118000
Influence
119000
Neutrality
122000
Create
124000
Content Key
126000
Security
128000
120

FIG. 4

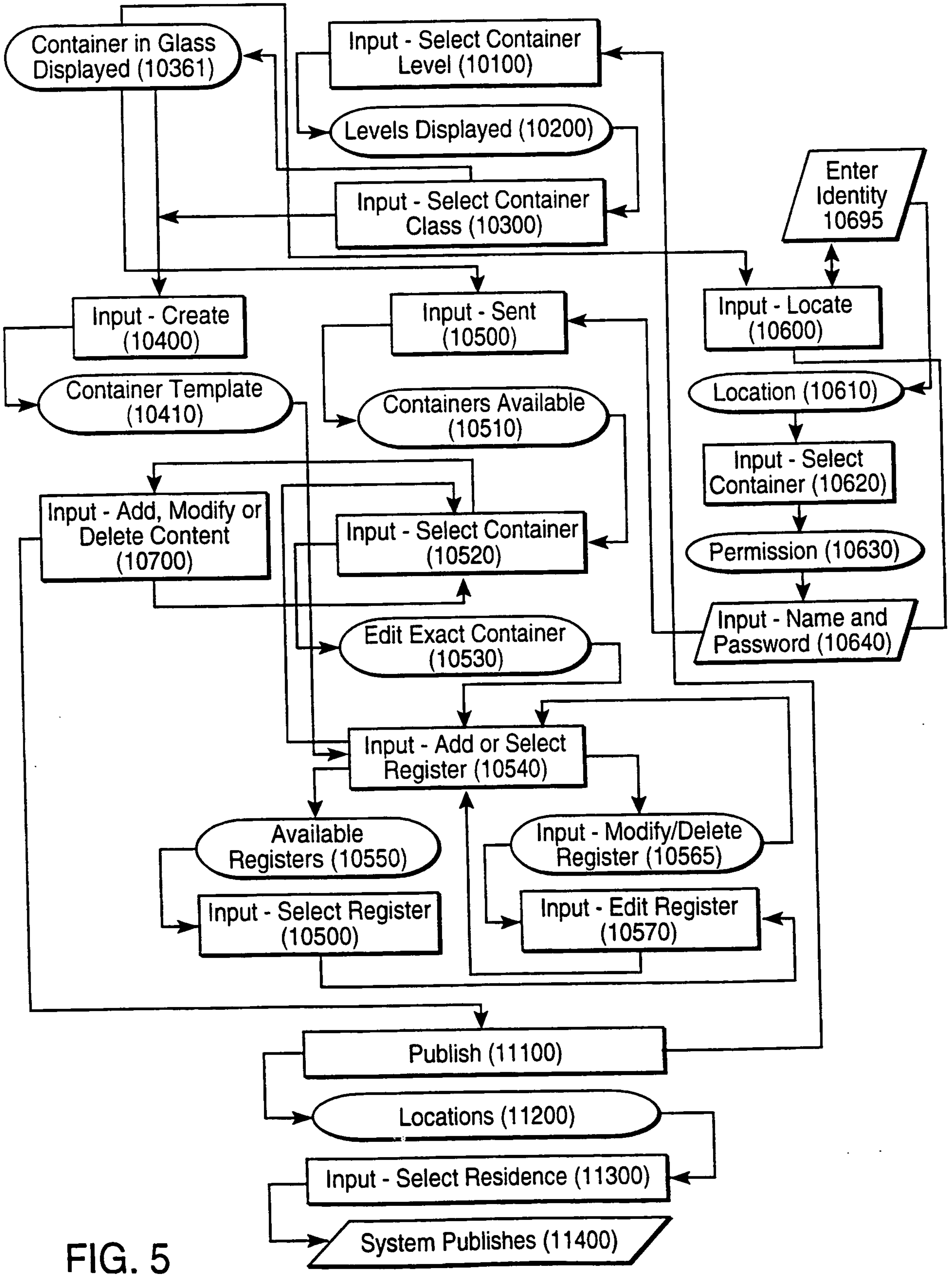

Container in Glass Displayed (10361)
Input - Select Container Level (10100)
Levels Displayed (10200)
Input - Select Container Class (10300)
Enter Identity 10695
Input - Create (10400)
Input - Sent (10500)
Input - Locate (10600)
Container Template (10410)
Containers Available (10510)
Location (10610)
Input - Select Container (10620)
Input - Add, Modify or Delete Content (10700)
Input - Select Container (10520)
Permission (10630)
Edit Exact Container (10530)
Input - Name and Password (10640)
Input - Add or Select Register (10540)
Available Registers (10550)
Input - Modify/Delete Register (10565)
Input - Select Register (10500)
Input - Edit Register (10570)
Publish (11100)
Locations (11200)
Input - Select Residence (11300)
System Publishes (11400)

FIG. 5

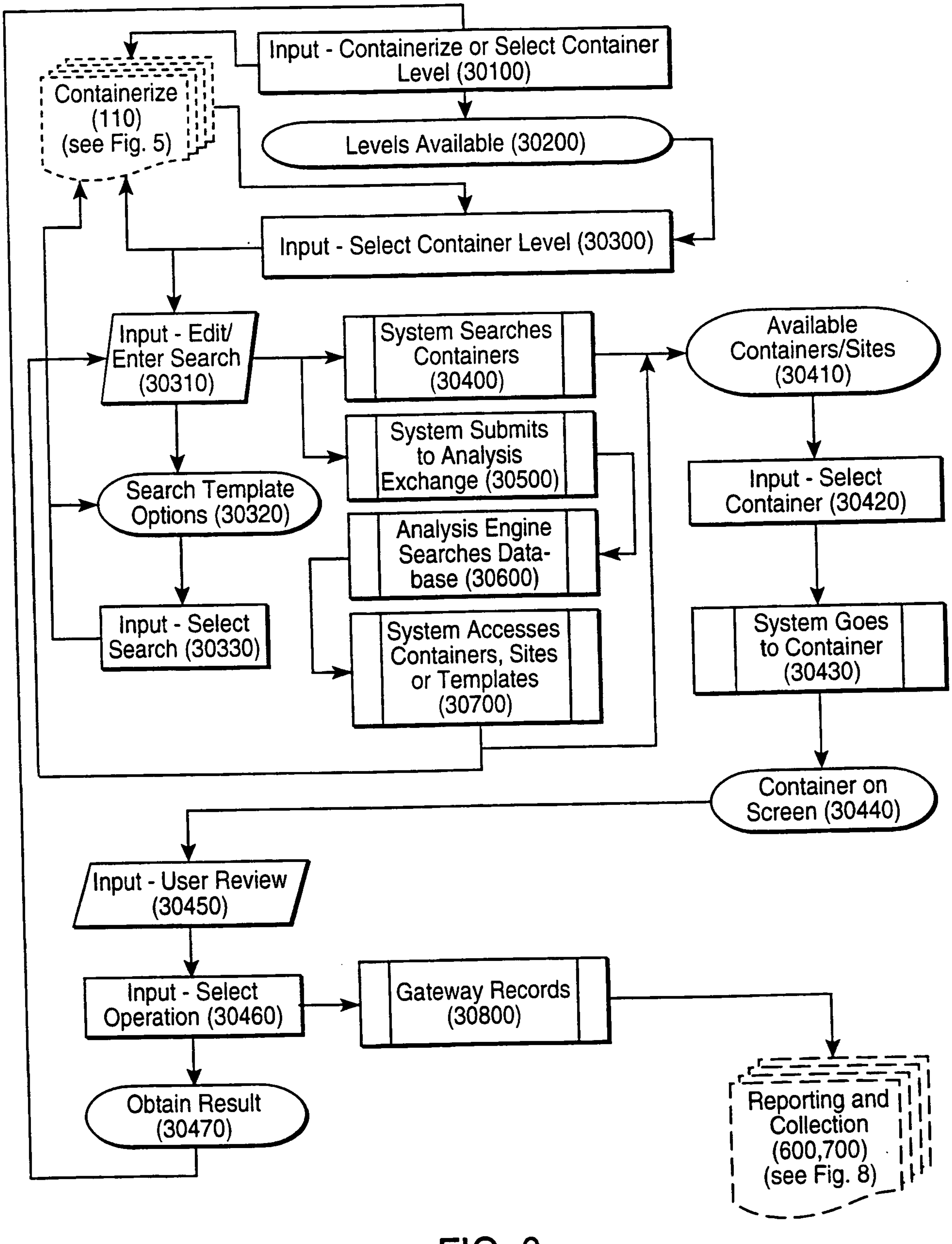

Input - Containerize or Select Container Level (30100)
Containerize (110) (see Fig. 5)
Levels Available (30200)
Input - Select Container Level (30300)
Input - Edit/ Enter Search (30310)
System Searches Containers (30400)
Available Containers/Sites (30410)
System Submits to Analysis Exchange (30500)
Search Template Options (30320)
Input - Select Container (30420)
Analysis Engine Searches Data-base (30600)
Input - Select Search (30330)
System Accesses Containers, Sites or Templates (30700)
System Goes to Container (30430)
Container on Screen (30440)
Input - User Review (30450)
Input - Select Operation (30460)
Gateway Records (30800)
Obtain Result (30470)
Reporting and Collection (600,700) (see Fig. 8)

FIG. 6

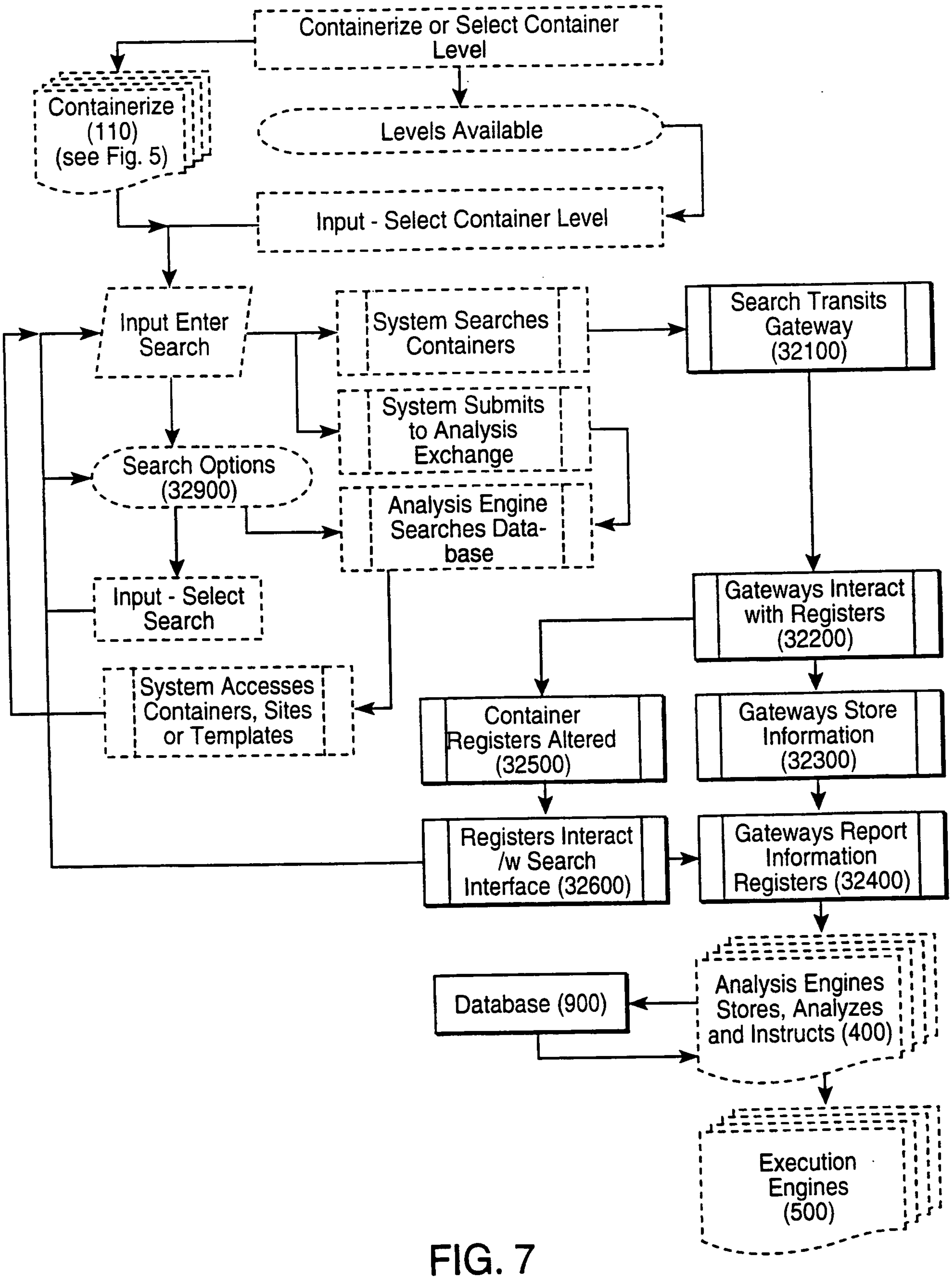

Containerize or Select Container Level
Containerize (110) (see Fig. 5)
Levels Available
Input - Select Container Level
Input Enter Search
System Searches Containers
Search Transits Gateway (32100)
System Submits to Analysis Exchange
Search Options (32900)
Analysis Engine Searches Database
Input - Select Search
Gateways Interact with Registers (32200)
System Accesses Containers, Sites or Templates
Container Registers Altered (32500)
Gateways Store Information (32300)
Registers Interact /w Search Interface (32600)
Gateways Report Information Registers (32400)
Database (900)
Analysis Engines Stores, Analyzes and Instructs (400)
Execution Engines (500)

FIG. 7

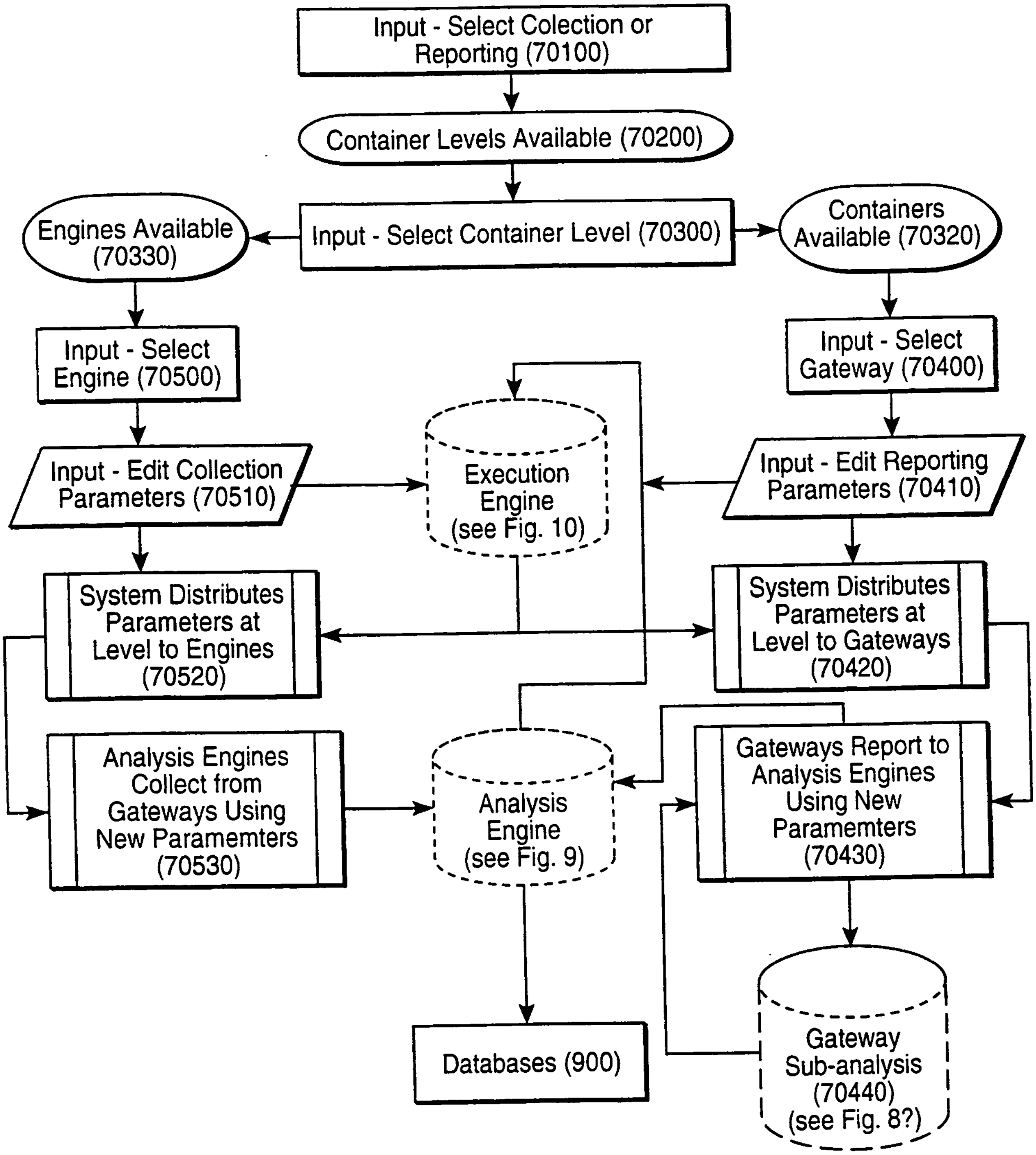

Input - Select Colection or Reporting (70100)
Container Levels Available (70200)
Input - Select Container Level (70300)
Engines Available (70330)
Containers Available (70320)
Input - Select Engine (70500)
Input - Select Gateway (70400)
Input - Edit Collection Parameters (70510)
Execution Engine (see Fig. 10)
Input - Edit Reporting Parameters (70410)
System Distributes Parameters at Level to Engines (70520)
System Distributes Parameters at Level to Gateways (70420)
Analysis Engines Collect from Gateways Using New Paramemters (70530)
Analysis Engine (see Fig. 9)
Gateways Report to Analysis Engines Using New Paramemters (70430)
Databases (900)
Gateway Sub-analysis (70440) (see Fig. 8?)

FIG. 8

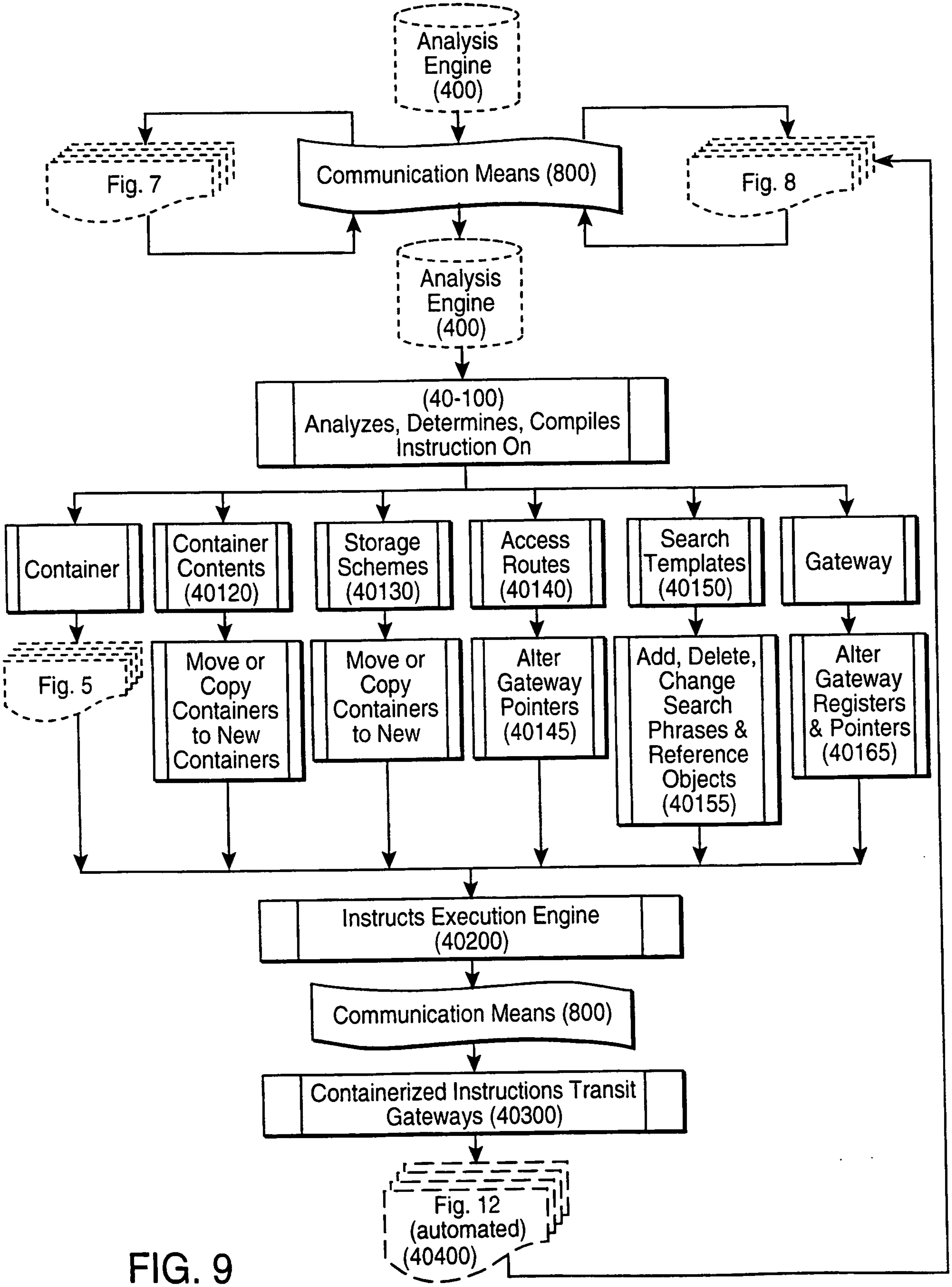

Analysis Engine (400)
Communication Means (800)
Fig. 7
Fig. 8
Analysis Engine (400)
(40-100) Analyzes, Determines, Compiles Instruction On
Container
Container Contents (40120)
Storage Schemes (40130)
Access Routes (40140)
Search Templates (40150)
Gateway
Fig. 5
Move or Copy Containers to New Containers
Move or Copy Containers to New
Alter Gateway Pointers (40145)
Add, Delete, Change Search Phrases & Reference Objects (40155)
Alter Gateway Registers & Pointers (40165)
Instructs Execution Engine (40200)
Communication Means (800)
Containerized Instructions Transit Gateways (40300)
Fig. 12 (automated) (40400)

FIG. 9

EXECUTION ENGINE

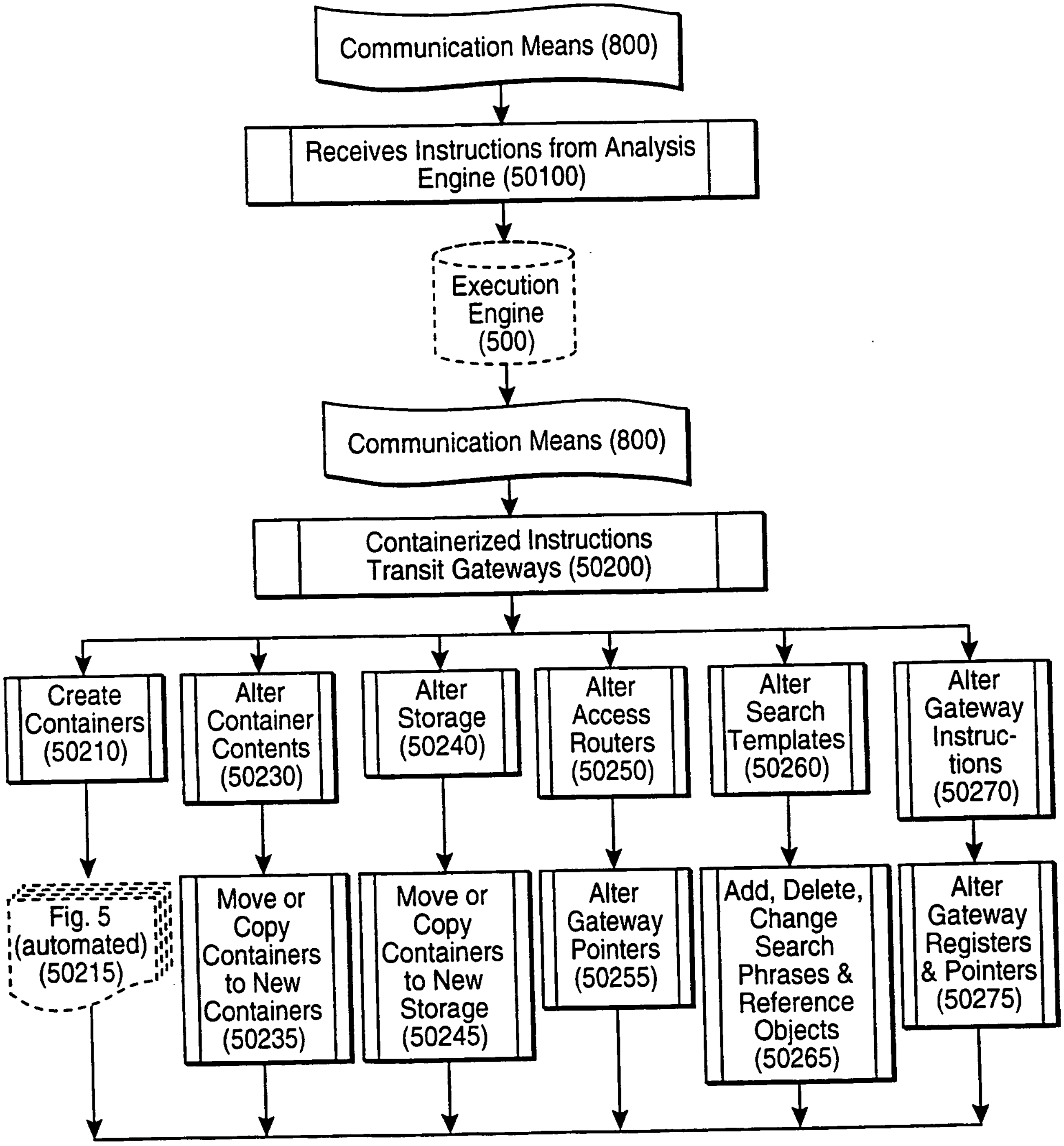

Communication Means (800)
Receives Instructions from Analysis Engine (50100)
Execution Engine (500)
Communication Means (800)
Containerized Instructions Transit Gateways (50200)
Create Containers (50210)
Alter Container Contents (50230)
Alter Storage (50240)
Alter Access Routers (50250)
Alter Search Templates (50260)
Alter Gateway Instruc-tions (50270)
Fig. 5 (automated) (50215)
Move or Copy Containers to New Containers (50235)
Move or Copy Containers to New Storage (50245)
Alter Gateway Pointers (50255)
Add, Delete, Change Search Phrases & Reference Objects (50265)
Alter Gateway Registers & Pointers (50275)

FIG. 10

GATEWAY EDITOR

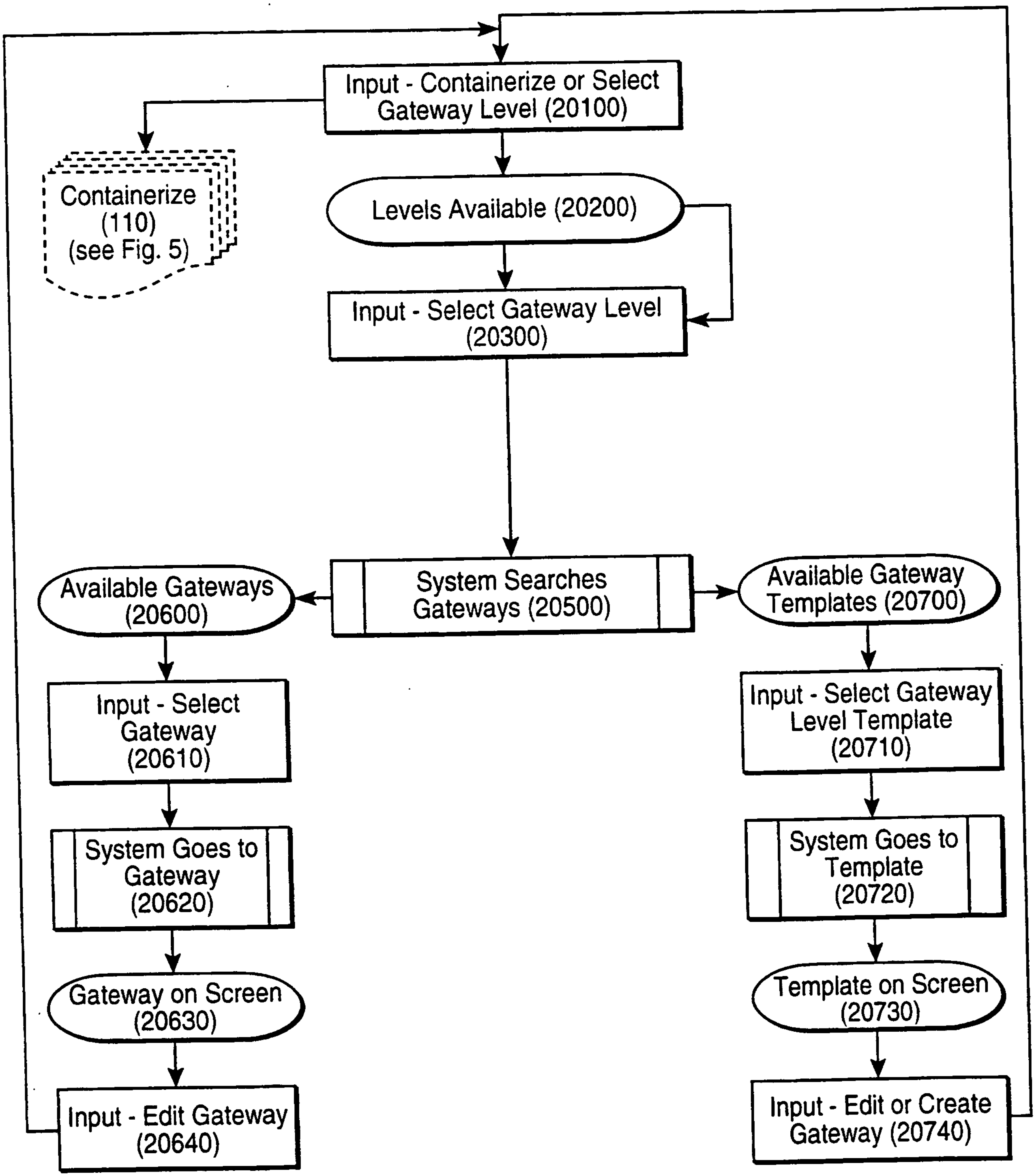

Input - Containerize or Select Gateway Level (20100)
Containerize (110) (see Fig. 5)
Levels Available (20200)
Input - Select Gateway Level (20300)
System Searches Gateways (20500)
Available Gateways (20600)
Available Gateway Templates (20700)
Input - Select Gateway (20610)
System Goes to Gateway (20620)
Gateway on Screen (20630)
Input - Edit Gateway (20640)
Input - Select Gateway Level Template (20710)
System Goes to Template (20720)
Template on Screen (20730)
Input - Edit or Create Gateway (20740)

FIG. 11

GATEWAY PROCESS

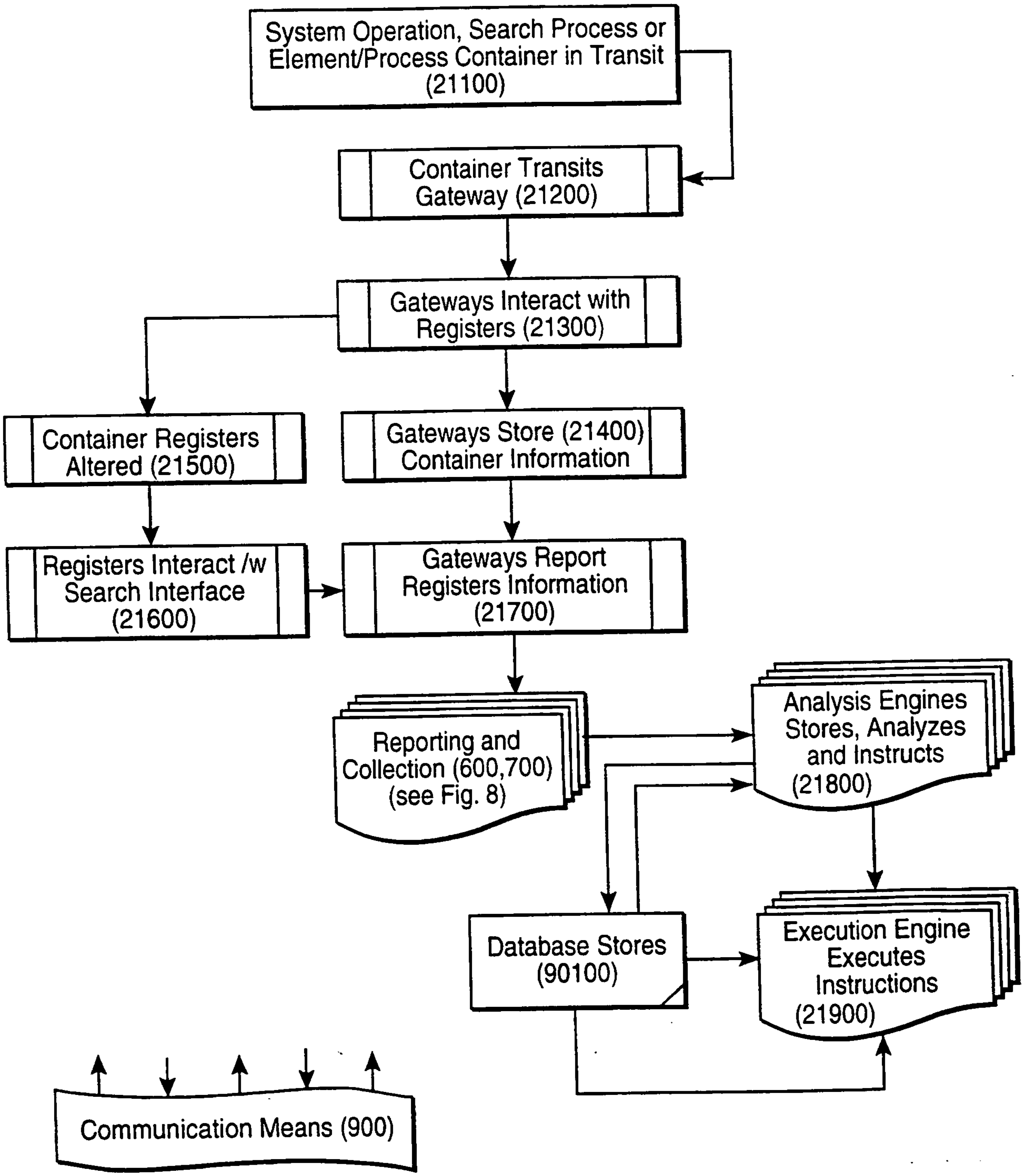

System Operation, Search Process or Element/Process Container in Transit (21100)
Container Transits Gateway (21200)
Gateways Interact with Registers (21300)
Container Registers Altered (21500)
Gateways Store (21400) Container Information
Registers Interact /w Search Interface (21600)
Gateways Report Registers Information (21700)
Reporting and Collection (600,700) (see Fig. 8)
Analysis Engines Stores, Analyzes and Instructs (21800)
Database Stores (90100)
Execution Engine Executes Instructions (21900)
Communication Means (900)

FIG. 12

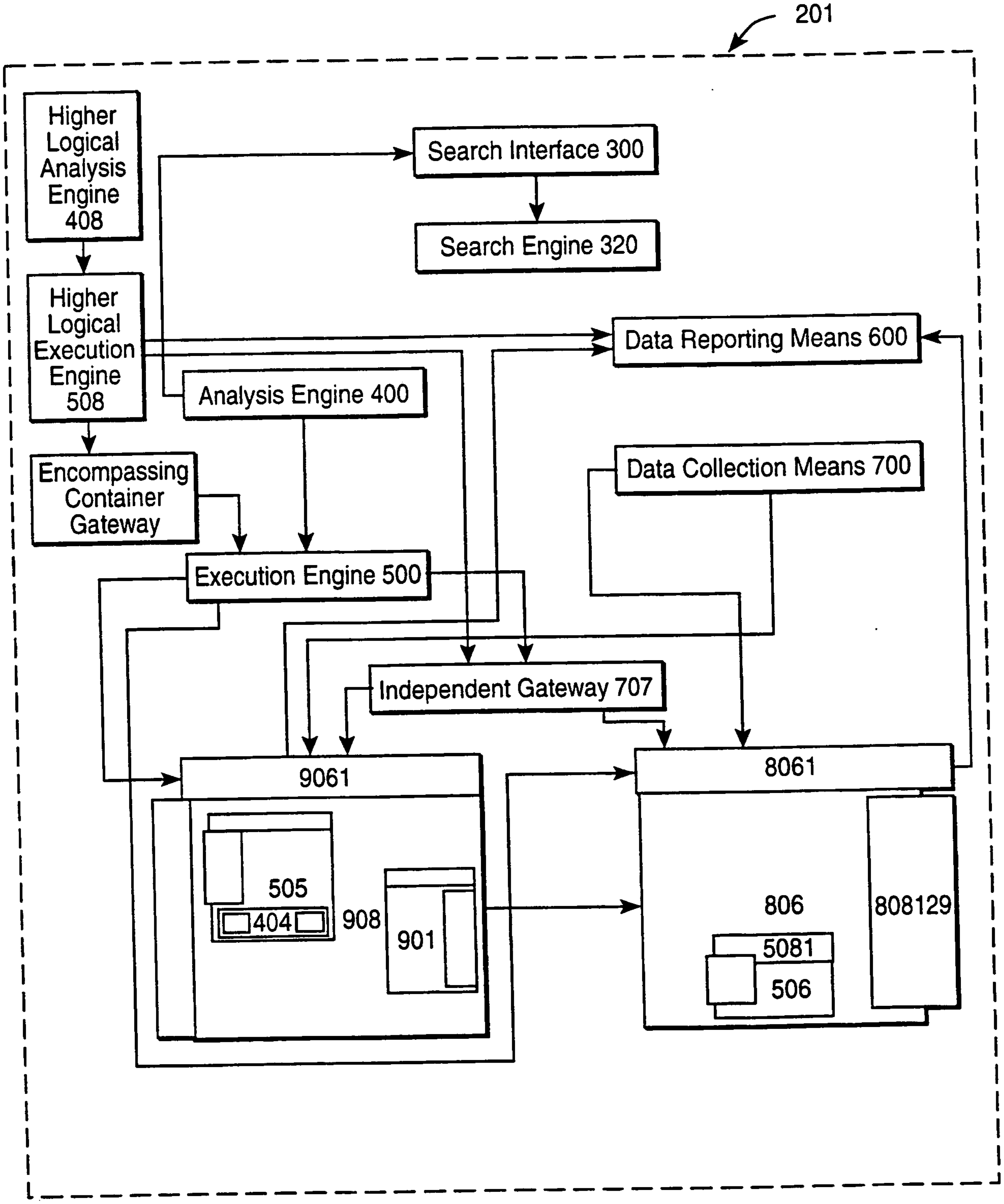

201
Higher Logical Analysis Engine 408
Search Interface 300
Search Engine 320
Higher Logical Execution Engine 508
Data Reporting Means 600
Analysis Engine 400
Data Collection Means 700
Encompassing Container Gateway
Execution Engine 500
Independent Gateway 707
9061
8061
505
404
908
901
806
808129
5081
506

FIG. 13A

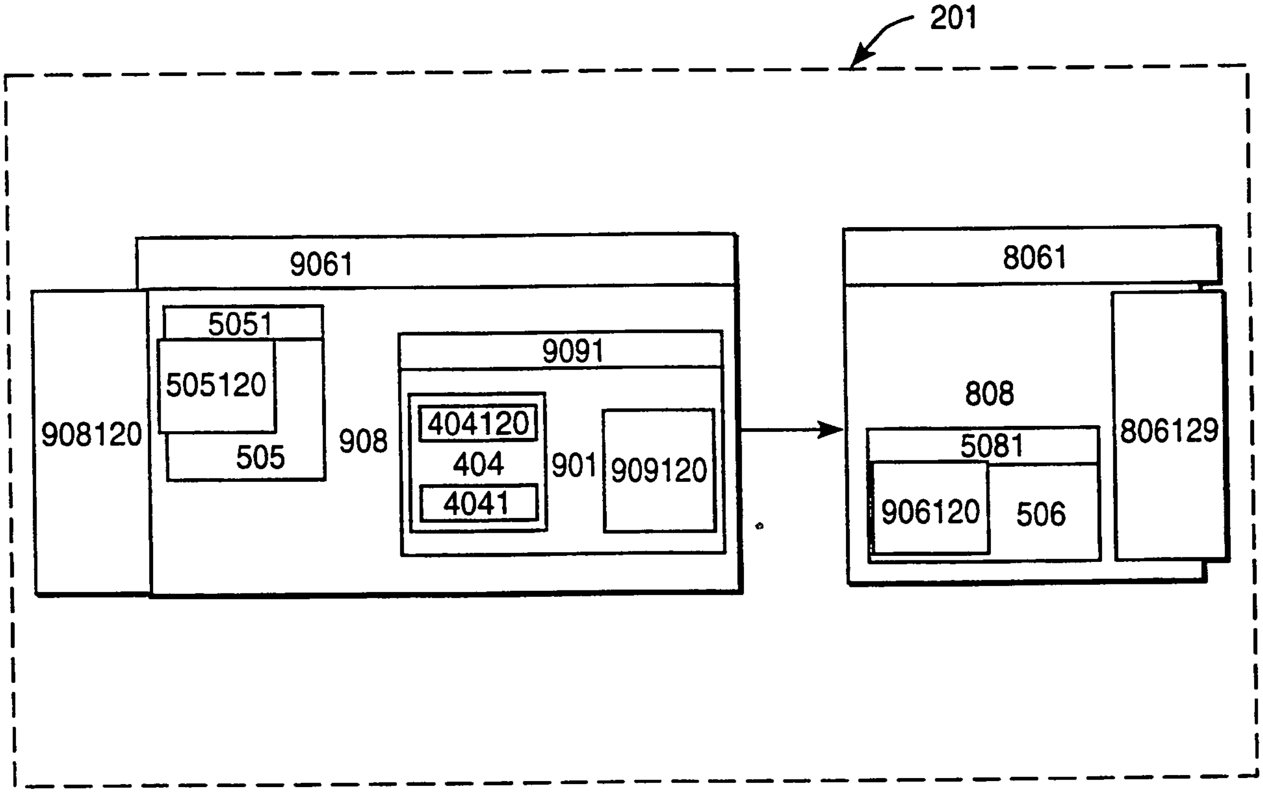

201
9061
908120
5051
505120
505
908
9091
404120
404
4041
901
909120
8061
808
5081
906120
506
806129

FIG. 13B

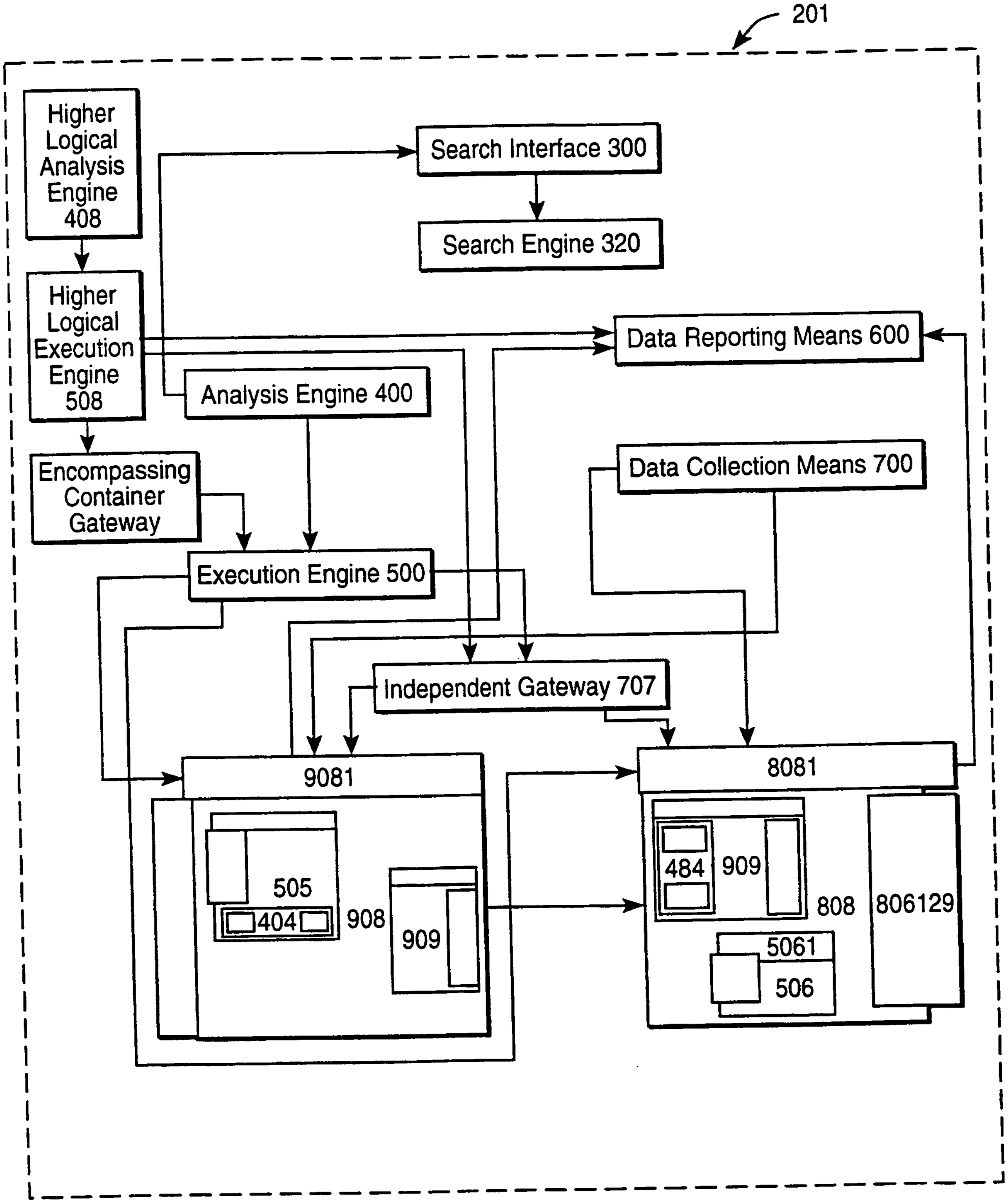

201
Higher Logical Analysis Engine 408
Search Interface 300
Search Engine 320
Higher Logical Execution Engine 508
Data Reporting Means 600
Analysis Engine 400
Encompassing Container Gateway
Data Collection Means 700
Execution Engine 500
Independent Gateway 707
9081
8081
505
404
908
909
484
909
808
806129
5061
506

FIG. 13C

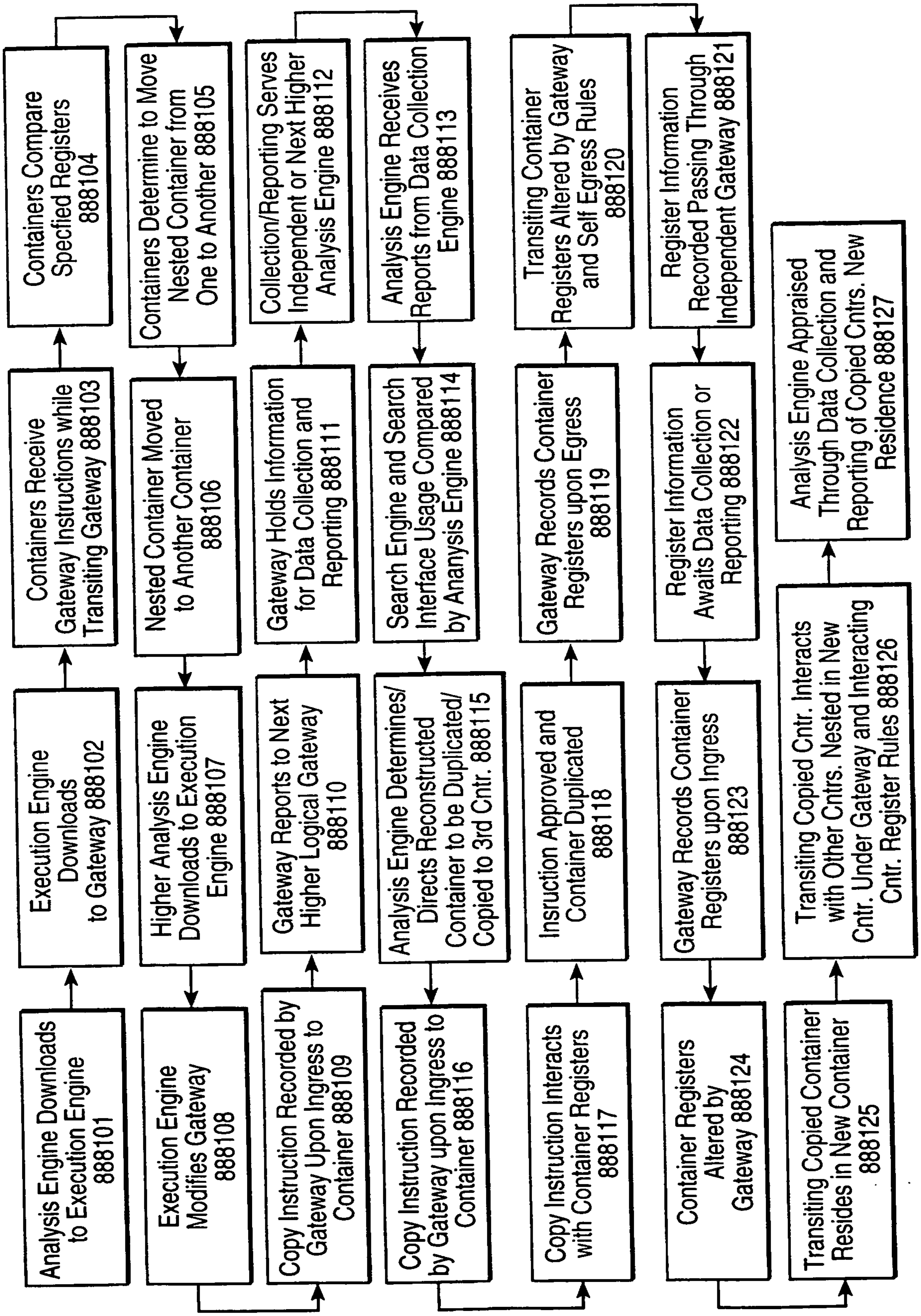

Analysis Engine Downloads to Execution Engine 888101
Execution Engine Downloads to Gateway 888102
Containers Receive Gateway Instructions while Transiting Gateway 888103
Containers Compare Specfied Registers 888104
Containers Determine to Move Nested Container from One to Another 888105
Nested Container Moved to Another Container 888106
Higher Analysis Engine Downloads to Execution Engine 888107
Execution Engine Modifies Gateway 888108
Copy Instruction Recorded by Gateway Upon Ingress to Container 888109
Gateway Reports to Next Higher Logical Gateway 888110
Gateway Holds Information for Data Collection and Reporting 888111
Collection/Reporting Serves Independent or Next Higher Analysis Engine 888112
Analysis Engine Receives Reports from Data Collection Engine 888113
Search Engine and Search Interface Usage Compared by Ananysis Engine 888114
Analysis Engine Determines/ Directs Reconstructed Container to be Duplicated/ Copied to 3rd Cntr. 888115
Copy Instruction Recorded by Gateway upon Ingress to Container 888116
Copy Instruction Interacts with Container Registers 888117
Insruction Approved and Container Duplicated 888118
Gateway Records Container Registers upon Egress 888119
Transiting Container Registers Altered by Gateway and Self Egress Rules 888120
Register Information Recorded Passing Through Independent Gateway 888121
Register Information Awaits Data Collection or Reporting 888122
Gateway Records Container Registers upon Ingress 888123
Container Registers Altered by Gateway 888124
Transiting Copied Container Resides in New Container 888125
Transiting Copied Cntr. Interacts with Other Cntrs. Nested in New Cntr. Under Gateway and Interacting Cntr. Register Rules 888126
Analysis Engine Appraised Through Data Collection and Reporting of Copied Cntrs. New Residence 888127

FIG. 13D

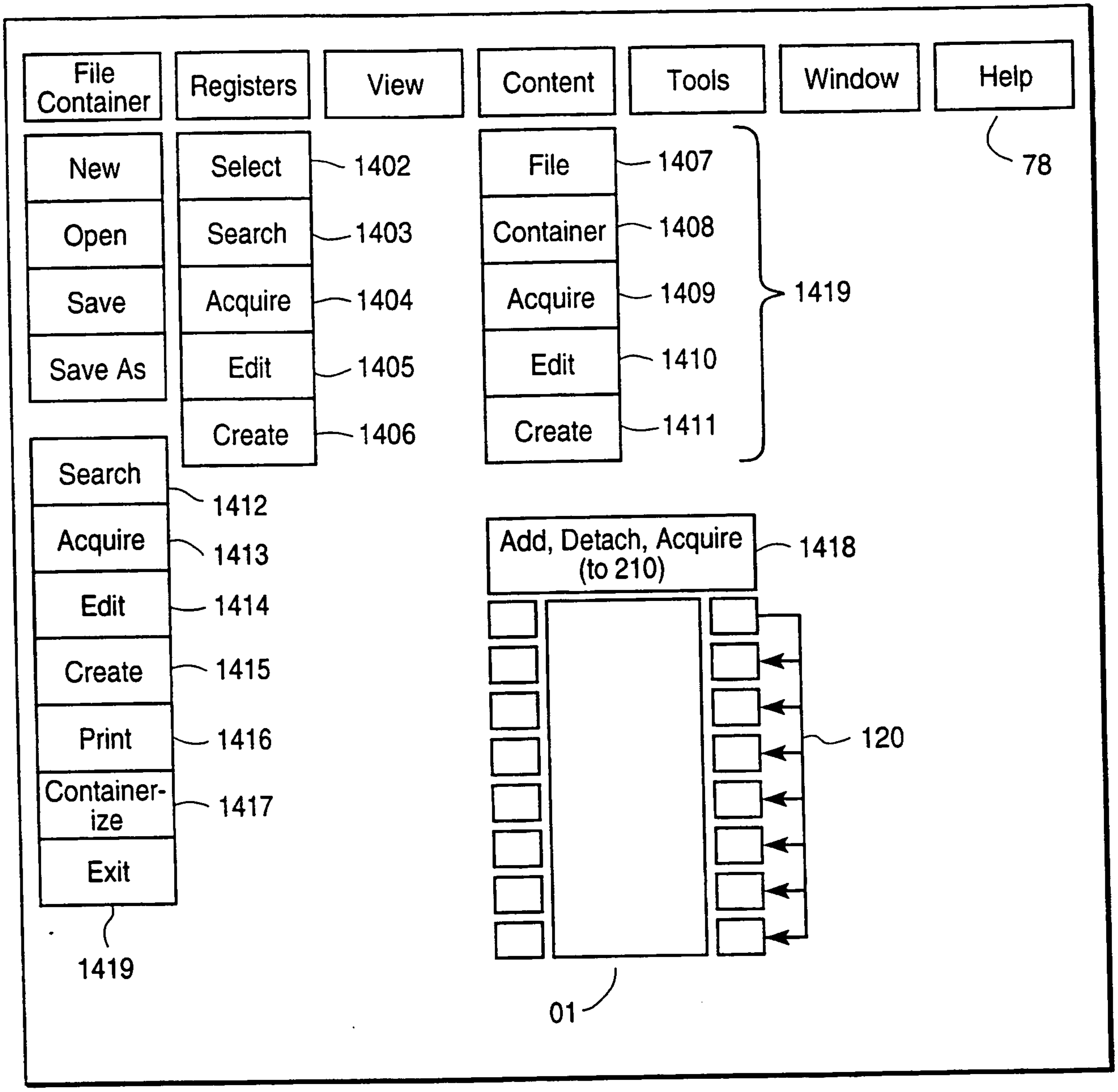

File Container
Registers
View
Content
Tools
Window
Help
78
New
Open
Save
Save As
Select
1402
Search
1403
Acquire
1404
Edit
1405
Create
1406
File
1407
Container
1408
Acquire
1409
Edit
1410
Create
1411
1419
Search
1412
Acquire
1413
Edit
1414
Create
1415
Print
1416
Container-ize
1417
Exit
1419
Add, Detach, Acquire (to 210)
1418
120
01

FIG. 14

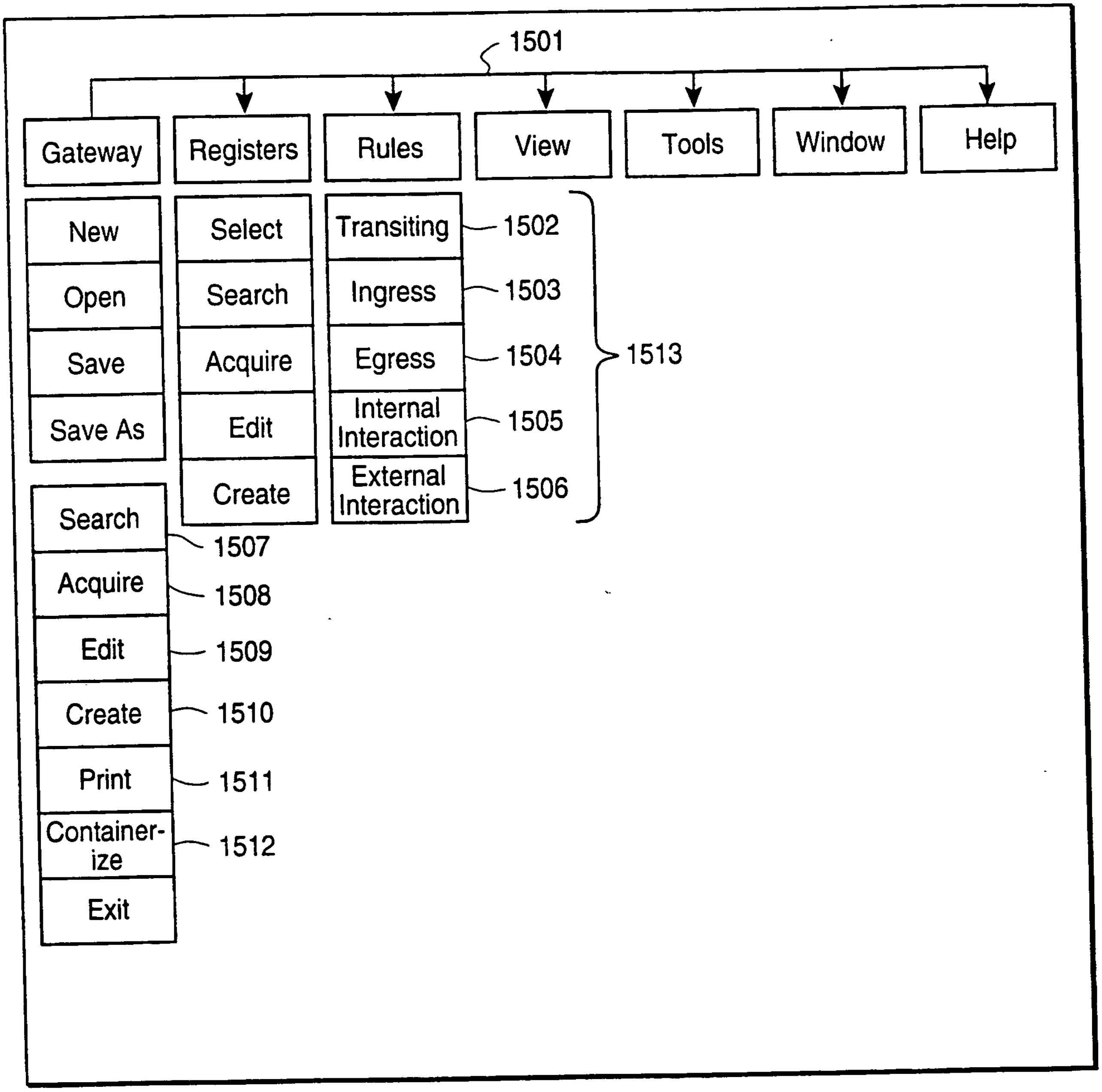

1501
Gateway
Registers
Rules
View
Tools
Window
Help
New
Open
Save
Save As
Select
Search
Acquire
Edit
Create
Transiting
Ingress
Egress
Internal Interaction
External Interaction
1502
1503
1504
1505
1506
1513
Search
Acquire
Edit
Create
Print
Container-ize
Exit
1507
1508
1509
1510
1511
1512

FIG. 15

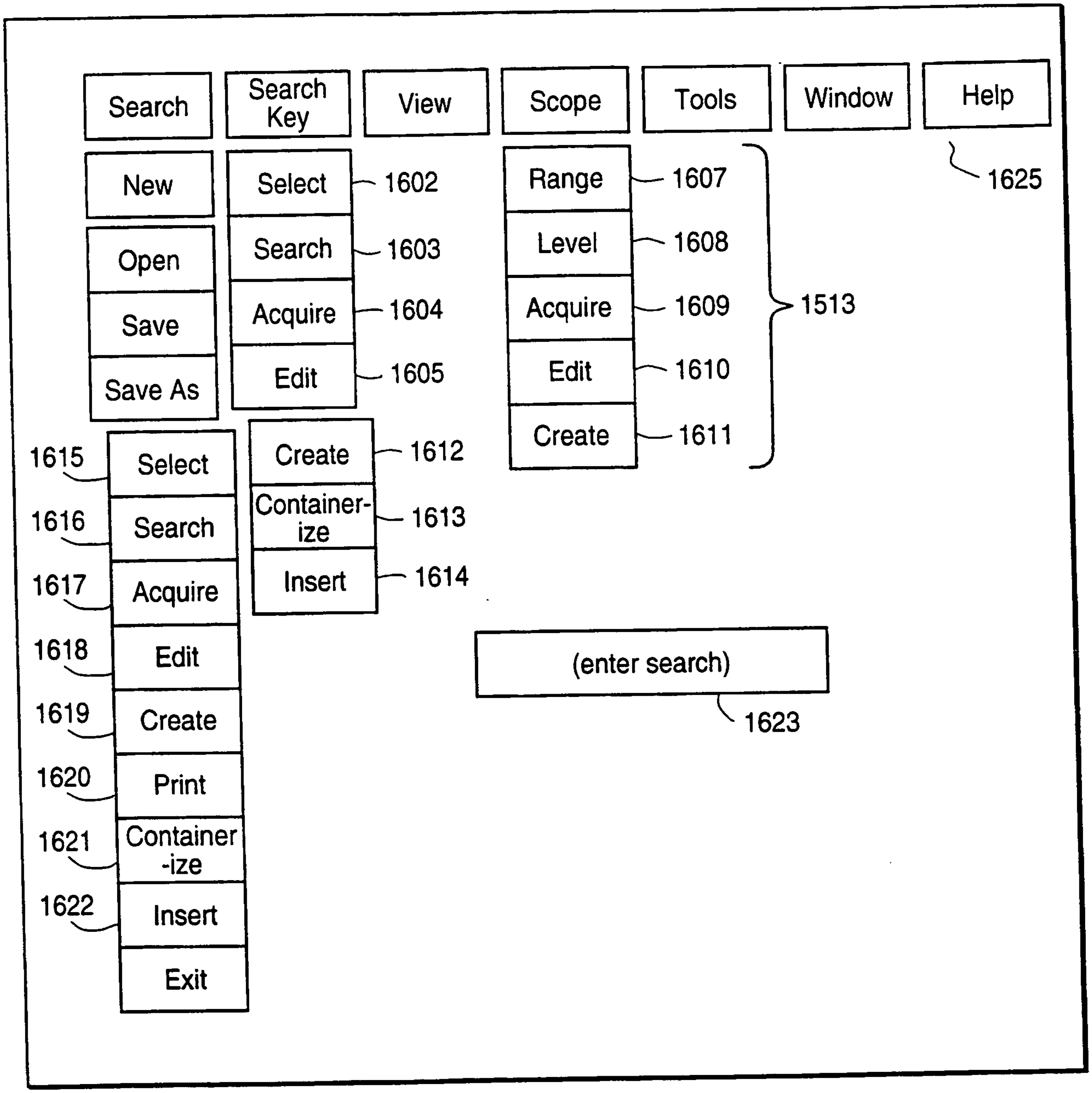

Search
Search Key
View
Scope
Tools
Window
Help
1625
New
Open
Save
Save As
Select
1602
Search
1603
Acquire
1604
Edit
1605
Range
1607
Level
1608
Acquire
1609
Edit
1610
Create
1611
1513
1615
Select
1616
Search
1617
Acquire
1618
Edit
1619
Create
1620
Print
1621
Container -ize
1622
Insert
Exit
Create
1612
Container-ize
1613
Insert
1614
(enter search)
1623

FIG. 16

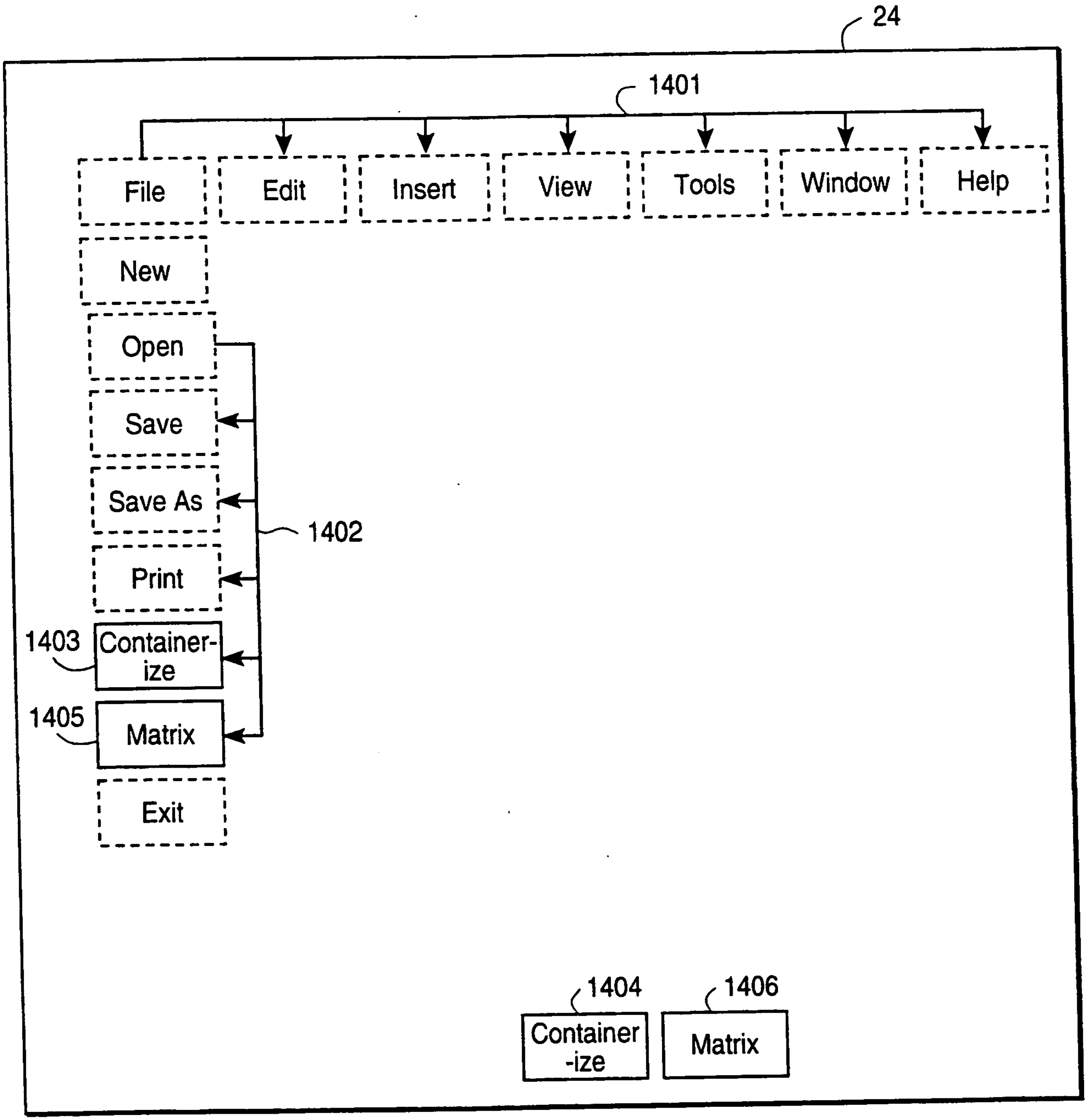

24
1401
File
Edit
Insert
View
Tools
Window
Help
New
Open
Save
Save As
1402
Print
1403
Container-ize
1405
Matrix
Exit
1404
Container-ize
1406
Matrix

FIG. 17

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/01988

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 17/30, 3/14
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : Please See Extra Sheet.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

MICROSOFT COMPUTER DICTIONARY

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, PRO-QUEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,768,510 A (GISH et al) 16 June 1998, column 5.	1-36
A	US 5,848,246 A (GISH et al) 08 December 1998, column 5.	1-36

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

03 JUNE 1999

Date of mailing of the international search report

15 JUN 1999

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer FOR
RUAY LIAN HO
Telephone No. (703) 305-3834

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/01988

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

707/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 101, 102, 103, 104, 200, 201, 202, 203, 204, 205, 206; 709/202, 203, 218, 228; 713/200, 201

B. FIELDS SEARCHED
Minimum documentation searched
Classification System: U.S.

707/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 101, 102, 103, 104, 200, 201, 202, 203, 204, 205, 206; 709/202, 203, 218, 228; 713/200, 201

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIBDATASHEET

Bib Data Sheet

CONFIRMATION NO. 1910

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
09/284,113	04/07/1999 **RULE**	707	2161	3726-US

APPLICANTS

MICHAEL DE ANGELO, SANTA BARBARA, CA;

** **CONTINUING DATA** *************************

This application is a 371 of PCT/US99/01988 01/28/1999
which claims benefit of 60/073,209 01/30/1998

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 04/12/2000

Foreign Priority claimed □ yes □ no
35 USC 119 (a-d) conditions met □ yes □ no □ Met after Allowance
Verified and Acknowledged ______________ Examiner's Signature Initials

STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
CA	30	36	3

ADDRESS

Fish & Richardson
500 Arguello Street
Sute 500
Redwood City ,CA 94063

TITLE

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

FILING FEE RECEIVED
669

FEES: Authority has been given in Paper
No. _________ to charge/credit DEPOSIT ACCOUNT
No. _________ for following:

□ All Fees
□ 1.16 Fees (Filing)
□ 1.17 Fees (Processing Ext. of time)
□ 1.18 Fees (Issue)
□ Other _______________
□ Credit

PATENT

9/20/00
#3

IN THE UNITED STATES

PATENT AND TRADEMARK OFFICE

APPLICANT:	Michael De Angelo
SERIAL NO.:	09/284,113
FILING DATE:	April 7, 1999
TITLE:	System And Method For Creating And Manipulating Information Containers With Dynamic Registers
EXAMINER:	Unknown
GROUP ART UNIT:	Unknown
ATTY. DKT. NO.:	3726

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner For Patents, Washington, D.C. 20231, on the date shown below:

Dated: July 7, 1999 By:
Greg T. Sueoka, Reg. No.: 33,800

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, DC. 20231

INFORMATION DISCLOSURE STATEMENT
Under 37 CFR §§ 1.56 and 1.97-98

SIR:

Pursuant to the provisions of 37 CFR §§ 1.56 and 1.97-98, enclosed herewith is modified form PTO-1449 listing references for consideration by the Examiner. A copy is enclosed herewith of each listed reference which may be material to the examination of this application, and with respect to which there may be a duty to disclose.

The filing of this Information Disclosure Statement shall not be construed as a representation regarding the completeness of the list of references, or that inclusion of a reference in this list is an admission that it is prior art or is pertinent to this application, or that a search has been made, or as an admission that the information listed is, or may be considered to be, material to patentability, or that no other material information exists, and shall not be construed as an admission against interest in any manner.

□ This application relies, under 35 U.S.C. § 120, on the earlier filing date of prior application Serial No. [SERIAL NUMBER], filed on [FILING DATE], and the references cited therein are hereby referenced, but are not required to be provided in this application under 37 CFR § 1.98(d).

The Information Disclosure Statement submitted herewith is being filed:

□ within three months of the filing date of the application, or date of entry into the national stage of an international application, or before the mailing date of a first official action on the merits, whichever event last occurred; OR

□ after three months of the filing date of this national application or the date of entry of the national stage in an international application, or after the mailing date of the first official action on the merits, whichever event last occurred, but before the mailing date of the first to occur of either:

(1) a final action under 37 CFR §1.113; OR

(2) a notice of allowance under 37 CFR §1.311; AND

□ attached hereto is the fee of $240, as set forth under 37 CFR §1.17(p), for submission of this Information Disclosure Statement under 37 CFR.§ 1.97(c); OR

☑ Applicant certifies pursuant to 37 CFR § 1.97(e) that:

☑ each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Statement; OR

□ no item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the knowledge of the person signing this certification after making reasonable inquiry, was known to any individual designated under 37 CFR § 1.56(c) more than three months prior to the filing of this Statement.

OR

□ before the payment of the issue fee but after the mailing date of the first to occur of either:

[1] a final action under 37 CFR § 1.113; OR

[2] a notice of allowance under 37 CFR § 1.311; AND
in accordance with the requirements of 37 CFR § 1.97(d):

□ Applicant certifies pursuant to 37 CFR. § 1.97(e) that:

□ each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Statement; OR

□ no item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the knowledge of the person signing this certification after making reasonable inquiry, was known to any individual designated under 37 CFR § 1.56(c) more than three months prior to the filing of this Statement; AND

□ Applicant hereby respectfully petitions for the consideration of the accompanying Information Disclosure Statement under 37 CFR § 1.97(d)(2); AND

□ Applicant submits the petition fee of $130 as set forth in 37 CFR § 1.17(i).

☑ Applicant submits that no fee is required for the consideration of the accompanying Information Disclosure Statement.

Consideration of the listed references and favorable action are solicited.

Respectfully submitted,
MICHAEL DE ANGELO

Dated: July 7, 1999

By: ________________
Greg T. Sueoka, Reg. No.: 33,800
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
Tel.: (650) 858-7194
Fax.: (650) 494-1417

3726
2771
0500

PTO/SB/21 (modified)
Approved for use through xx/xx/xx, OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

0001/PTO
Rev. 10/95

U.S. Department of Commerce
Patent and Trademark Office

TRANSMITTAL FORM

(to be used for all correspondence during pendency of filed application)

Application Number	09/284,113
Filing Date	April 7, 1999
First Named Inventor	Michael De Angelo
Group Art Unit Number	Unknown
Examiner Name	Unknown
Attorney Docket Number	3726

Total Number of Pages in This Submission: *5

ENCLOSURES *(check all that apply)*

□ Fee Transmittal Form (in duplicate)
  □ Check Enclosed
☑ Return Receipt Postcard
□ Response to Notice to File Missing Parts
□ Assignment & Recordation Cover Sheet
□ Declaration
□ Small Entity Statement
☑ Information Disclosure Statement & PTO-1449
  ☑ Copies of IDS Cited References
□ Request for Corrected Filing Receipt
□ Request for Correction of Recorded Assignment
□ Amendment/Response: [ ] Page(s)
  □ After Final
□ Status Request
□ Revocation and Power of Attorney

□ Issue Fee Transmittal
□ Letter to Chief Draftsperson
□ Formal Drawing(s):
  [ ] Sheet(s) of Figure(s) [ ]
□ Appeal Communication to Board of Appeals and Interferences
□ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
□ Certified Copy of Priority Document(s)
□ After Allowance Communication to Group
□ ______
□ ______
□ ______
□ ______
□ ______
□ ______

REMARKS: *Does not include total pages of cited references

SIGNATURE OF ATTORNEY OR AGENT

Signature:	Greg Sue		
Attorney/Reg. No.:	Greg T. Sueoka / Reg. No.: 33,800	Dated:	July 7, 1999

CERTIFICATE OF MAILING

I hereby certify that this correspondence, including the enclosures identified above, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: The Assistant Commissioner for Patents, Washington, D.C. 20231 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.

Signature:	Greg Sue		
Typed or Printed Name:	Greg T. Sueoka	Dated:	July 7, 1999
Express Mail Mailing Number (*optional*):			

Rev. 07/06/99

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

09/284113

U.S. APPLICATION NO.	FIRST NAMED APPLICANT	ATTY. DOCKET NO.
09/284,113	DE ANGELO	M 3726 US

INTERNATIONAL APPLICATION NO	
PCT/US99/01988	
I.A. FILING DATE	PRIORITY DATE
01/28/99	01/30/98

5611

GREG T SUEOKA
FENWICK & WEST
TWO PALO ALTO SQUARE
PALO ALTO CA 94306

DATE MAILED: 01/13/00

NOTIFICATION OF ACCEPTANCE OF APPLICATION UNDER 35 U.S.C. 371 AND 37 CFR 1.494 OR 1.495

1. The applicant is hereby advised that the United States Patent and Trademark Office in its capacity as ☑ a Designated Office (37 CFR 1.494), □ an Elected Office (37 CFR 1.495), has determined that the above identified international application has met the requirements of 35 U.S.C. 371, and is **ACCEPTED** for national patentability examination in the United States Patent and Trademark Office.

2. The United States Application Number assigned to the application is shown above and the relevant dates are:

April 7, 1999
35 U.S.C. 102(e) DATE

April 7, 1999
DATE OF RECEIPT OF
35 U.S.C. 371 REQUIREMENTS

A Filing Receipt (PTO-103X) will be issued for the present application in due course. **THE DATE APPEARING ON THE FILING RECEIPT AS THE "FILING DATE" IS THE DATE ON WHICH THE LAST OF THE 35 U.S.C. 371(C) REQUIREMENTS HAS BEEN RECEIVED IN THE OFFICE. THIS DATE IS SHOWN ABOVE.** *The filing date of the above identified application is the international filing date of the international application (Article 11(3) and 35 U.S.C. 363).* Once the Filing Receipt has been received, send all correspondence to the Group Art Unit designated thereon.

3. ☑ A request for immediate examination under 35 U.S.C. 371(f) was received on April 7, 1999 and the application will be examined in turn.

4. The following items have been received:

☑ U.S. Basic National Fee.
☑ Copy of the international application in:
  □ a non-English language.
  ☑ English.
□ Translation of the international application into English.
☑ Oath or Declaration of inventors(s) for DO/EO/US.
□ Copy of Article 19 amendments. □ Translation of Article 19 amendments into English.
  The Article 19 amendments □ have □ have not been entered.
□ The International Preliminary Examination Report in English and its Annexes, if any.
□ Copy of the Annexes to the International Preliminary Examination Report (IPER).
  □ Translation of Annexes to the IPER into English.
 The Annexes □ have □ have not been entered.
□ Preliminary amendment(s) filed ____________ and ____________.
□ Information Disclosure Statement(s) filed____________ and ____________.
☑ Assignment document.
☑ Power of Attorney and/or Change of Address.
□ Substitute specification filed ____________.
☑ Statement Claiming Small Entity Status.
□ Priority Document.
☑ Copy of the International Search Report □ and copies of the references cited therein.
□ Other:

Applicant is reminded that any communication to the United States Patent and Trademark Office must be mailed to the address given in the heading and include the U.S. application no. shown above. (37 CFR 1.5)

Barbara Campbell
National Stage Processing
Telephone: (703) (703) 305-3631

FORM PCT/DO/EO/903 (December 1997)

PC #4 Receipt
CFR

PTO/SB/21 (modified)
Approved for use through xx/xx/xx, OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

0001/PTO Rev. 10/95	U.S. Department of Commerce Patent and Trademark Office	Application Number	09/284,113
		Filing Date	April 7, 1999
TRANSMITTAL FORM *(to be used for all correspondence during pendency of filed application)*		First Named Inventor	Michael De Angelo
		Group Art Unit Number	2771
		Examiner Name	not yet known
Total Number of Pages in This Submission	5	Attorney Docket Number	3726

ENCLOSURES *(check all that apply)*

- □ Fee Transmittal Form (in duplicate)
  - □ Check Enclosed
- ☒ Return Receipt Postcard
- □ Response to Notice to File Missing Parts
- □ Assignment & Recordation Cover Sheet
- □ Declaration
- □ Small Entity Statement
- □ Information Disclosure Statement & PTO-1449
  - □ Copies of IDS Cited References
- ☒ Request for Corrected Filing Receipt
- □ Request for Correction of Recorded Assignment
- □ Amendment/Response: [ ] Page(s)
  - □ After Final
- □ Status Request
- □ Revocation and Power of Attorney

- □ Issue Fee Transmittal
- □ Letter to Chief Draftsperson
- □ Formal Drawing(s):
  - [ ] Sheet(s) of Figure(s) [ ]
- □ Appeal Communication to Board of Appeals and Interferences
- □ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- □ Certified Copy of Priority Document(s)
- □ After Allowance Communication to Group
- ☒ Copy of Official Filing Receipt
- ☒ Copy of executed Verified Statement Claiming Small Entity Status
- □ ______
- □ ______
- □ ______
- □ ______

REMARKS:

SIGNATURE OF ATTORNEY OR AGENT

Signature:	Greg Sueoka		
Attorney/Reg. No.:	Greg T. Sueoka / Reg. No.: 33,800	Dated:	May 8, 2000

CERTIFICATE OF MAILING

I hereby certify that this correspondence, including the enclosures identified above, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: The Assistant Commissioner for Patents, Washington, D.C. 20231 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.

Signature:	Greg Sueoka		
Typed or Printed Name:	Greg T. Sueoka	Dated:	May 8, 2000
Express Mail Mailing Number *(optional)*:			

IN THE UNITED STATES

PATENT AND TRADEMARK OFFICE

APPLICANT: Michael De Angelo

SERIAL NO.: 09/284,113

FILING DATE: April 7, 1999

TITLE: System And Method For Creating And Manipulating Information Containers With Dynamic Registers

EXAMINER: not yet known

GROUP ART UNIT: 2771

ATTY. DKT. NO.: 3726

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner For Patents, Washington, D.C. 20231, on the date shown below:

Dated: May 8 2000

By:
Greg T. Sueoka, Reg. No.: 33,800

ASSISTANT COMMISSIONER FOR PATENTS
APPLICATION PROCESSING DIVISION
CUSTOMER CORRECTION BRANCH
WASHINGTON, DC. 20231

REQUEST FOR CORRECTED FILING RECEIPT

SIR:

Enclosed is a copy of the Official Filing Receipt. It contains the following error:

1. The filing receipt does not indicate small entity status, as evidenced by the executed Verified Statement Claiming Small Entity Status (37 CFR 1.9(f) & 1.27(c))—Small Business Concern, a copy of which is enclosed.

Please issue a corrected Filing Receipt rectifying this error.

☑ The correction is not due to any error by the Applicant and therefore no fee is due.

□ Since at least one of the corrections is due to Applicant's error, payment in the amount of $25, pursuant to 37 CFR § 1.19(h), is enclosed.

Respectfully submitted,
MICHAEL DE ANGELO

Dated: May 8, 2000

By: ______________________
Greg T. Sueoka, Reg. No.: 33,800
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
Tel.: (650) 858-7194
Fax.: (650) 494-1417

21114/03726/DOCS/1042815.1

FILING RECEIPT

OC000000005048164

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: ASSISTANT SECRETARY AND
COMMISSIONER OF PATENT AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	DRAWINGS	TOT CLAIMS	IND CLAIMS
09/284,113	04/07/1999	2771	524	3726-US	30	36	3

GREG T SUEOKA
FENWICK & WEST
TWO PALO ALTO SQUARE
PALO ALTO, CA 94306

Date Mailed: 04/13/2000

Receipt is acknowledged of this nonprovisional Patent Application. It will be considered in its order and you will be notified as to the results of the examination. Be sure to provide the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION when inquiring about this application. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please write to the Office of Initial Patent Examination's Customer Service Center. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the PTO processes the reply to the Notice, the PTO will generate another Filing Receipt incorporating the requested corrections (if appropriate).**

Applicant(s)

MICHAEL DE ANGELO, SANTA BARBARA, CA UNITED STATES;

Small Entity Status

Continuing Data as Claimed by Applicant

THIS APPLICATION IS A 371 OF PCT/US99/01988 01/28/1999
WHICH CLAIMS BENEFIT OF 60/073,209 01/30/1998

Foreign Applications

If Required, Foreign Filing License Granted 04/12/2000

**

Title

SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

Preliminary Class

707

Data entry by : BARRETO, NGA **Team :** OIPE **Date:** 04/13/2000

PTO/SB/10 (6-95) (modified)
Approved for use through 07/31/96. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) ~~& 1.27(c)~~)–SMALL BUSINESS CONCERN	Docket Number (Optional): 3726

Applicant or Patentee: Michael De Angelo

Application or Patent No.: ______

Filing Date or Issue Date: ______

Title: System And Method For Creating And Manipulating Information Containers With Dynamic Registers

I hereby declare that I am

[ ] the owner of the small business concern identified below:

[X] an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN Ematrix Corporation

ADDRESS OF SMALL BUSINESS CONCERN 104 West Anapamu, Suite C
Santa Barbara, California 93101

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

[X] the specification filed herewith with title as listed above.

[ ] the application identified above.

[ ] the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each such person, concern or organization having any rights in the invention is listed below:

[X] No such person, concern, or organization exists.

[ ] Each such person, concern or organization is listed below:

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING Michael De Angelo

TITLE OF PERSON IF OTHER THAN OWNER Officer

ADDRESS OF PERSON SIGNING 104 West Anapamu, Suite C, Santa Barbara, California 93101

SIGNATURE ______ DATE April 15, 1999

Rev. 04/01/99

PATENT #5 4-10-01

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

APPLICANT:	Michael De Angelo
APPLICATION NO.:	09/284,113
FILING DATE:	April 7, 1999
TITLE:	System And Method For Creating And Manipulating Information Containers With Dynamic Registers
EXAMINER:	Not yet assigned
GROUP ART UNIT:	2771
ATTY. DKT. NO.:	3726

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner For Patents, Washington, D.C. 20231 on the date shown below:

Dated: 3/29/01 By:
Greg T. Sueoka, Reg. No.: 33,800

COMMISSIONER FOR PATENTS
WASHINGTON, DC. 20231

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT
Under 37 CFR §§ 1.56 and 1.97-98

SIR:

Pursuant to the provisions of 37 CFR §§ 1.56 and 1.97-98, enclosed herewith is modified form PTO-1449 listing references for consideration by the Examiner. A copy is enclosed herewith of each listed reference, which may be material to the examination of this application, and with respect to which there may be a duty to disclose.

The filing of this Information Disclosure Statement shall not be construed as a representation regarding the completeness of the list of references, or that inclusion of a reference in this list is an admission that it is prior art or is pertinent to this application, or that a search has been made, or as an admission that the information listed is, or may be considered to be, material to patentability, or that no other material information exists, and shall not be construed as an admission against interest in any manner.

□ This application relies, under 35 U.S.C. § 120, on the earlier filing date of prior application Serial No. [SERIAL NUMBER], filed on [FILING DATE], and the references cited therein are hereby referenced, but are not required to be provided in this application under 37 CFR § 1.98(d).

The Information Disclosure Statement submitted herewith is being filed:

□ within three months of the filing date of the application, or date of entry into the national stage of an international application, or before the mailing date of a first official action on the merits, whichever event last occurred; OR

□ before the mailing of a first official action after the filing of a request for continued examination (RCE) under 37 CFR § 1.114;

□ after three months of the filing date of this national application or the date of entry of the national stage in an international application, or after the mailing date of the first official action on the merits, whichever event last occurred, but before the mailing date of the first to occur of either:

(1) a final action under 37 CFR §1.113; OR

(2) a notice of allowance under 37 CFR §1.311; AND

□ attached hereto is the fee of $180, as set forth under 37 CFR §1.17(p), for submission of this Information Disclosure Statement under 37 CFR.§ 1.97(c); OR

☒ Applicant certifies pursuant to 37 CFR § 1.97(e) that:

☒ each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Statement; OR

□ no item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the knowledge of the person signing this certification after making reasonable inquiry, was known to any individual designated under 37 CFR § 1.56(c) more than three months prior to the filing of this Statement.

OR

□ before the payment of the issue fee but after the mailing date of the first to occur of either:

[1] a final action under 37 CFR § 1.113; OR

[2] a notice of allowance under 37 CFR § 1.311; AND
in accordance with the requirements of 37 CFR § 1.97(d):

□ Applicant certifies pursuant to 37 CFR. § 1.97(e) that:

□ each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Statement; OR

□ no item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the knowledge of the person signing this certification after making reasonable inquiry, was known to any individual designated under 37 CFR § 1.56(c) more than three months prior to the filing of this Statement; AND

□ Applicant hereby respectfully petitions for the consideration of the accompanying Information Disclosure Statement under 37 CFR § 1.97(d)(2); AND

□ Applicant submits the petition fee of $180 as set forth in 37 CFR § 1.17(p).

☒ Applicant submits that no fee is required for the consideration of the accompanying Information Disclosure Statement.

Consideration of the listed references and favorable action are solicited.

Respectfully submitted,
MICHAEL DE ANGELO

Dated: 3/29/01

By: Greg Sueoka
Greg T. Sueoka, Reg. No.: 33,800
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
Tel.: (650) 858-7194
Fax.: (650) 494-1417

~~2771~~
+
2171

PTO/SB/21 (modified)
Approved for use through xx/xx/xx, OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

0001/PTO
Rev. 10/95

U.S. Department of Commerce
Patent and Trademark Office

TRANSMITTAL FORM

(to be used for all correspondence during pendency of filed application)

Application Number	09/284,113
Filing Date	April 7, 1999
First Named Inventor	Michael De Angelo
Group Art Unit Number	2771
Examiner Name	Not yet assigned
Attorney Docket Number	3726
Total Number of Pages in This Submission	*5

ENCLOSURES *(check all that apply)*

- □ Fee Transmittal Form (in duplicate)
  - □ Check Enclosed
- ☒ Return Receipt Postcard
- □ Response to Notice to File Missing Parts
- □ Assignment & Recordation Cover Sheet
- □ Declaration
- □ Power of Attorney
- □ Application Data Sheet
- ☒ Information Disclosure Statement & PTO-1449
  - ☒ Copies of IDS Cited References
- □ Request for Corrected Filing Receipt
- □ Request for Correction of Recorded Assignment
- □ Amendment/Response: [ ] Page(s)
  - □ After Final
- □ Status Request
- □ Revocation and Substitute Power of Attorney

- □ Issue Fee Transmittal
- □ Letter to Chief Draftsperson
- □ Formal Drawing(s):
  - [ ] Sheet(s) of Figure(s) [ ]
- □ Appeal Communication to Board of Appeals and Interferences
- □ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- □ Certified Copy of Priority Document(s)
- □ After Allowance Communication to Group
- □ ____________
- □ ____________
- □ ____________
- □ ____________
- □ ____________
- □ ____________

REMARKS: *Does not include total number of pages of the cited references

SIGNATURE OF ATTORNEY OR AGENT

Signature:			
Attorney/Reg. No.:	Greg T. Sueoka / Reg. No.: 33,800	Dated:	3/29/01

CERTIFICATE OF MAILING

I hereby certify that this correspondence, including the enclosures identified above, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.

Signature:			
Typed or Printed Name:	Greg T. Sueoka	Dated:	3/27/01
Express Mail Mailing Number *(optional)*:			

D

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **G06F 17/30**	**A1**	**(11) International Publication Number:** **WO 98/02831** **(43) International Publication Date:** 22 January 1998 (22.01.98)

(21) International Application Number: PCT/US97/11885

(22) International Filing Date: 10 July 1997 (10.07.97)

(30) Priority Data:
08/678,680 11 July 1996 (11.07.96) US

(71) Applicant: TANDEM COMPUTERS INCORPORATED [US/US]; 10435 N. Tantau Avenue, Loc 200-16, Cupertino, CA 95014 (US).

(72) Inventors: ERLENKOETTER, Ansgar; Auf der Heide 44, D-61267 Neu-Anspach (DE). DE BORST, Jeroen, Peter; Alte Mauergasse 5, D-61348 Bad Homberg (US). BONHAM, Peter, Douglas; Am Alten Bach 19, D-61352 Bad Homberg (DE).

(74) Agents: GRANATELLI, Lawrence, W. et al.; Graham & James L.L.P., 600 Hansen Way, Palo Alto, CA 94304 (US).

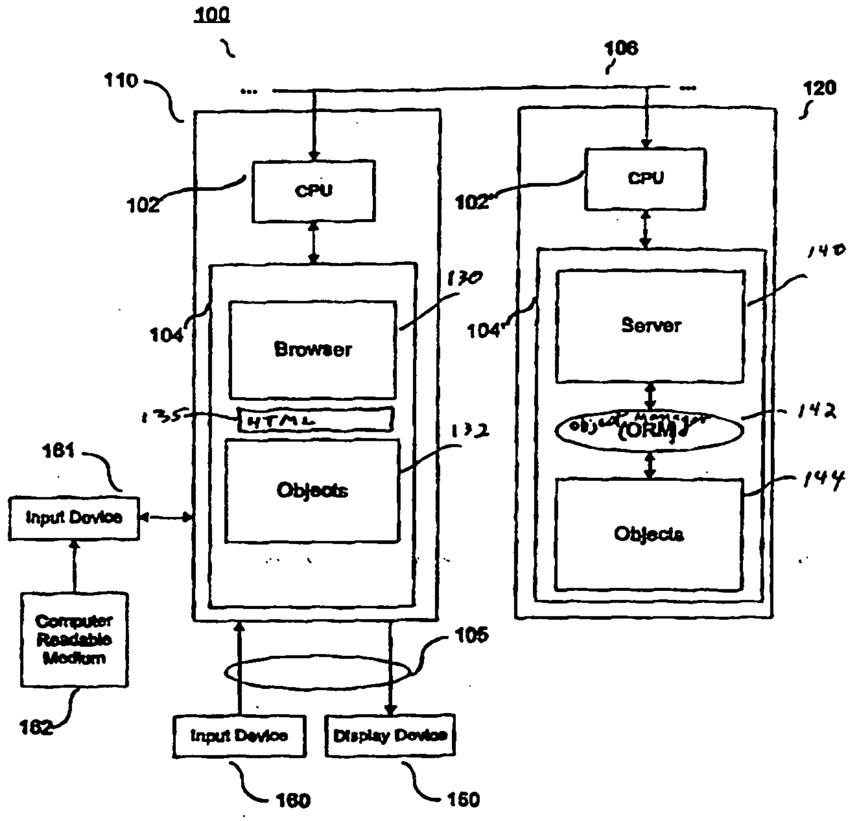
(81) Designated States: JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: HYPERMEDIA OBJECT MANAGEMENT

(57) Abstract

A method and apparatus that uses a hypermedia approach to managing distributed objects. A first embodiment of the present invention uses the World Wide Web hypermedia system. A user initializes browser software that allows the user to browse and change various attributes of objects in the system. The browser communicates with a server that includes an http adapter and a gateway. The gateway can access objects in the system and generate HTML code in accordance with the objects. One embodiment of the present invention uses hierarchical tree-oriented objects. These objects are "self-describing" (also called "introspective"). The server queries the objects in response to the queries from the browser and each queried object responds with information about itself. In another preferred embodiment, the server initiates queries of the objects and retains this information for use in responding to later queries from the browser.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

HYPERMEDIA OBJECT MANAGEMENT

FIELD OF THE INVENTION

This application relates to object oriented programming and, in particular, to management of distributed objects via the World Wide Web.

BACKGROUND OF THE INVENTION

The past several years have seen an explosive growth of the use of distributed objects. Now, a single system may be composed of objects obtained from different vendors and having different interfaces. Such objects are called "heterogeneous objects." Thus, a system can be formed of a large and rapidly changing number of heterogeneous objects. Such a system requires a flexible and adaptive approach for system and application management. Conventionally, a heterogeneous system is managed by way of object-specific presentation facilities, i.e., by way of a user front-end that was written for each type of heterogeneous object. Such an approach is, however, too expensive in both development time and maintenance and administrative costs. In addition, conventional object management is often achieved through a single management center. Use of a single center is not efficient when a large number of objects need to be managed.

SUMMARY OF THE INVENTION

The present invention overcomes the problems and disadvantages of the prior art by using a hypermedia approach to object management. In this approach, each object is akin to a hypermedia document. The described embodiment of the present invention uses the World Wide Web hypermedia system. In a preferred embodiment of the present invention, a user initializes browser software that allows the user to browse and change various attributes of objects in the system. The browser communicates with a server that includes an http adapter and a gateway. The gateway can access objects in the system and generate HTML code in accordance with the objects.

A described embodiment of the present invention uses hierarchical tree-oriented objects. In a first embodiment, these objects are "self-describing" (also called "introspective"). The server queries the objects in response to the queries from the browser and each queried object responds with information about itself. In another preferred embodiment, the server initiates queries of the objects and retains this information for use in responding to later queries from the browser.

In accordance with the purpose of the invention, as embodied and broadly described herein the invention is a system for managing objects, including a first server, comprising: a first receiver portion configured to receive a request in a hypermedia format; a first translator portion configured to convert the hypermedia request to an object request;a sender portion configured to send the object request to an object manager; a second receiver portion configured to receive a response from the object manager; and a second translator portion configured to convert the object manager response to the hypermedia format.

In further accordance with the purpose of this invention, as embodied and broadly described herein the invention is a method for browsing objects, where a browser communicates with a server, comprising the steps, performed by the browser, of: sending an initial URL to the server; receiving first data from the server, where the first data specifies an object corresponding to the URL; sending user-entered data associated with the object to the server; and receiving second data from the server, where the second data specifies a second object corresponding to the user-entered data.

Advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a block diagram of a computer system in accordance with a preferred embodiment of the present invention.

Fig. 2 is another block diagram of a computer system in accordance with a preferred embodiment of the present invention.

Fig. 3 is a diagram of data sent between a browser, server, and object manager in accordance with the embodiment of Fig. 1.

Fig. 4 is a diagram of a format in which objects are organized.

Fig. 5 shows another example of a page displayed by the browser.

Figs. 6(a) and 6(b) show an example of HTML that causes the browser to display a portion of the page of Fig. 5.

Figs. 7(a) through 7(c) show further examples of HTML that result in the portions of page of Fig. 5.

Figs. 8(a) and 8(b) show several examples of ORM (Object Resource Management) requests made by the server to the object manager and the resulting responses from the object manager.

Fig. 9 shows another page displayed by the browser.

Fig. 10 shows layers of functions available to the object manager.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Reference will now be made in detail to a preferred embodiment of the invention, an example of which is illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

I. System Overview

Fig. 1 is a block diagram of a computer system 100 in accordance with a preferred embodiment of the present invention. Computer system 100 includes a first computer 110 and a second computer 120. First computer 110 and second computer 120 are connected together via line 106, which can be, for example, a LAN, a WAN, or an internet connection. Line 106 can also represent a wireless connection, such as a cellular network connection.

First computer 110 includes a CPU 102; a memory 104; input/output lines 105; an input device 160, such as a keyboard or mouse; and a display device 150, such as a display terminal. First computer 110 also includes an input device 161 that reads computer instructions stored on computer readable medium 162. These instructions are the instructions of e.g., browser software 130. Memory 104 of first computer 110 includes browser software 130, Hypertext Markup Language (HTML) 135, and objects 132. A person of ordinary skill in the art will understand that memory 104 also contains additional information, such as application programs, operating systems, data, etc., which are not shown in the figure for the sake of clarity.

Second computer 120 includes a CPU 102' and a memory 104'. Memory 104' of second computer 120 includes server software 140, an object manager (ORM) 142, and objects 144. HTML 135 in the memory of first computer 110 was downloaded over line 106 from server 140 of second computer 120. A person of ordinary skill in the art will understand that memory 104' also contains additional information, such as application programs, operating systems, data, etc., which are not shown in the figure for the sake of clarity. Server 140, object manager 142, and objects 144 can also be located in memory 104 of first computer 110.

It will be understood by a person of ordinary skill in the art that computer system 100 can also include numerous elements not shown in the Figure for the sake of clarity, such as disk drives, keyboards, display devices, network connections, additional memory, additional CPUs, LANs, input/output lines, etc.

The following paragraphs provide a general discussion of the World Wide Web ("the Web"). The Web is built around a network of "server" computers, such as second computer 120, which exchange requests and data with each other using the hypertext transfer protocol ("http"). A human designer designs the layout of a Web page, which is then specified using HTML ("Hypertext Markup Language"). Several versions of HTML are currently in existence. Examples include HTML versions 2.0 and 3.0, as specified by the WWW Consortium of MIT. The HTML used in the described embodiment of the invention includes frames, forms, and tables, as are known to persons of ordinary skill in the art.

A user views a Web page using one of a number of commercially available "browser" programs. The browser submits an appropriate http request to establish a communications link with a Web server of the network. A typical http request references a Web page by its unique Uniform Resource Locator ("URL"). A URL identifies the Web server hosting that Web page, so that an http request for access to the Web page can be routed to the appropriate Web server for handling. Web pages can also be linked graphically to each other.

Fig. 2 is an additional block diagram of a computer system in accordance with a preferred embodiment of the present invention. Browser 130 communicates with server 140. Server 140 includes an http adapter 202 and a management gateway 204. Http Adapter 202 handles communication via the known http protocol. Management gateway 204 communicates with object manager 142. Server 140 communicates with one or more objects 132, 144 using a request/response (RR) protocol, such as the ORM (Object Resource Management) protocol, which is discussed below. Note that objects 132 and 144 can be located on the same or different physical computers or machines. Server 140 also communicates with external interface 206, which communicates using the known SNMP and CMIP protocols. Server 140 also communicates with external gateway 208, which communicates using the known SNMP and CMIP protocols. The system can contain more than one servers 140 and more objects than are shown in Fig. 4.

Fig. 3 is a diagram of data sent between a browser, server, and object manager in accordance with the embodiment of Fig. 1. In the example of Fig. 3, the user has already begun execution of browser software 130. In step 302, the user enters the URL of server 140 by way of browser 130. The browser sends a request to the server and, in step 304, the server responds with the HTML to generate a home page. The home page allows the user to enter a URL (or to chose a URL from those known provided within the HTML of the home page). The user can then chose to set/browse objects in the system, as described below. The user can also request information and statistics about once or more objects in the system.

In step 306, the user enters a URL of an object by way of browser 130. Server 140 converts the URL to a request to an object manager. For example, in the described embodiment, server 140 converts the URL to an ORM request, as described below. The ORM request is sent to the object manager, which returns object data in steps 308 and 310. Server 140 converts the object data into HTML, which is sent to browser 130 in step 312. The HTML may be based on a predetermined page template known to the server. Alternately, the format of a page may be determined "on the fly" based on the information obtained from the object manager. Server 140 converts all pathnames, such as object-links in the object data (see Fig. 4) to URLs in HTML and vice versa. Thus, if a user clicks on an area in a page displayed by the browser that corresponds to an object-link, browser 130 has the URL corresponding to that object-link. This new URL is sent to the server, which obtains the page information and sends HTML to display information for the object connected to the object-link.

Steps 314 through 320 represent a "set" mode, in which the user enters new values for an object by way of browser 130. In step 314, the user indicates that he wishes to enter "set" mode. This indication is usually accomplished by clicking on a button in the current page (thus, the HTML generated by server 140 should include HTML for this button). In step 316, server 140 sends a "form" for set mode. In step 318, the user enters new values into the form and clicks on "submit" (or "apply", (see Fig. 5), as is known to persons of ordinary skill in the art. Server 140 converts the submitted form to, for example, an ORM request, as described below. The ORM request is sent to the object manager, which returns object data in steps 317 and 319. Server 140 converts the object data of step 319 into HTML, which is sent to browser 130 in step 320.

Steps 322 through 332 represent a "browse" mode, in which the user views values associated with an object by way of browser 130. In step 322, the user indicates that he wishes to enter "browse" mode. This indication is usually accomplished by clicking on a button in the current page (thus, the HTML generated by server 140 should include HTML for this button). In step 324, server 140 sends a "form" for browse mode. In step 326, the user enters new values into the form and clicks on "submit" (or "apply", see Fig. 5), as is known to persons of ordinary skill in the art. Server 140 converts the submitted form to, for example, an ORM request, as described below. The ORM request is sent to the object manager, which returns data corresponding to the object in steps 328 and 330. Server 140 converts the response of step 330 into HTML, which is sent to browser 130 in step 332.

II. Hypermedia Object Management

A. Object Organization

Fig. 4 is a diagram of a format in which objects are organized in a preferred embodiment. This organization is transparent to server 140 and browser 130. It will be understood that the present invention can be used with a number of object organizations and with a number of object management protocols. The embodiment described herein uses the ORM protocol, as described below.

The model of Fig. 4 assumes the following:

1) Management operations can be mapped to two basic operations: a) Get an attribute (or a set of attributes) of an object and b) set an attribute (or set of attributes) of an object.

2) All entities to be managed can be organized as a directed tree with nodes and leaves where the nodes are either (callable) objects or components (sub-parts of objects) with attributes as the leaves (with combined name/pair values), and

3) All knowledge about management operations and attributes is built into and controlled by the managed object.

Fig. 4 shows the following types of entities:

1) Objects

Objects encapsulate and control management aspects and respective management operations. In the described embodiment, an object is identified by a "pathname," which is the destination for object calls. Each manageable object has its own virtual tree of components, attributes, and object-links.

2) Components

Components are the primary structuring mechanism within an object. Component sub-trees may be of arbitrary depth and component nodes may contain any number of object-links, other sub-components, or attributes.

3) Attributes

Attributes describe specific aspects of a component within an object (for example, "status=running" describes the state of a resource). Attribute nodes have additional properties beyond name and value, such s access mode and data type. Attribute nodes are leaves and do not have children.

4) Object-links

Object-links contain an object reference to a related object. As every object is responsible for its own virtual tree of resources, one object can provide a reference (hyperlink) to another object. Thus, in the described embodiment, a first object can have links to a second object, so that objects can be "walked" by way of browser 130.

5) Relations

Objects and components are the primary means for structuring and navigation in the described embodiment. Attributes have values that characterize the state of the resource. All operations (browsing and attribute retrieval/setting) are performed with respect to a single level of the tree (e.g., relative to a specific parent).

Server 140 preferably issues the following requests to object manager 142:

1) Get a list of linked objects,

2) Get a list of components and/or sub-components,

3) Get a list of attributes,

4) Set a list of attributes (Along with name/value pairs for each attribute), and

5) Get an extended list of attributes, which returns meta-information about the attribute, such as data type, allowed access mode (ro, rw) or valid ranges of new attribute values. Within the ORM model, all management operations are mapped to these five operations. Thus, every managed object preferably supports these five operations.

It should be understood that the attributes and object types shown in the examples herein are included only for the purposes of example. The present invention can be practiced using any appropriate object organization and type.

B. Server Interface

In the described embodiment, all messages passing in and out of server 140 are ASCII messages.

A example URL for object 402 of Fig. 4 would look like:

Http://ham/get/objectRoot/Component1/Component2/

A example URL for attribute 404 of Fig. 4 would look like:

Http://ham/get/objectRoot/Component1/Component2/Attr1/

In both of these URLs, "ham" stands for "HyperMedia Adapter to Management" and represents the address of server 140; "get" (this could also be "set") represents an operation to be performed on an object or attribute; and the remainder of the URL represents the tree of the object or attribute known to the object manager. Other URLs may also include additional information use, for example, by the object manager.

Fig. 5 shows a page displayed by browser 130 in "set" mode. Fig. 5 shows the values of attributes for a "Configuration" object component. These attributes include:

1) Status 520,

2) Maximum Concurrency 523,

3) Trace Level 524,

4) OSL Traces Enabled 526,

5) Script directory/Vol. 528,

6) Script File 530,

7) Cache Tcl Scripts 532,

8) Tcl Trace Enabled 534, and

9) Maximum Size of Synthesized Page 536.

Fig. 5 also shows an entry 522 for changing the status attribute. It should be understood that the attributes of Fig. 5 are presented for the sake of example only and are not to be taken in a limiting sense. Fig. 5 also shows a reset button 540 and an apply button 550. When the user clicks reset button, original attribute values are returned. When apply button 550 is clicked, browser 130 posts a form, as is known to persons of ordinary skill in the art.

Figs. 6(a) and 6(b) show an example of HTML generated by server 140. When browser 130 interprets the HTML of Fig. 6, it generates the portion containing attribute values 520-536 and buttons 540, 550 of Fig. 5. Figs. 7(a) through 7(c) show an example of HTML generated by server 140. When browser 130 interprets the HTML 702, 704, and 706 of Figs. 7(a) through 7(c), it generates portions 502, 506, and 504, respectively, of Fig. 5.

Fig. 9 shows another page displayed by browser 130 in accordance with HTML generated by server 140. The page of Fig. 9 is used to browse objects, but cannot change the attributes of objects.

The previous paragraphs discuss the browser GUI presented to the user and how server 140 translates between HTML and a protocol understood by the object manager. The following paragraphs describe the protocol used to communicate with object manager 142 about objects and to change objects in accordance with the HTML received by the server.

Figs. 8(a) and 8(b) show several examples of ORM requests made by the server 140 to object manager 142 and the resulting responses from object manager 142. Pages of the description shows formats of such requests and responses. Request 802 is an example of an OrmGet request sent from server 140 to object manager 142. The format of an OrmGet request is:

OrmGet: pathname
 entity types

where pathname is a name of an object or an attribute. Possible entity types are: "Object" (all known objects at this level), "Component" (a list of all components below the level of the path specified in the OrmGet), "Attribute" (a list of attributes for the current node; for every attribute, its name and "stringified value is returned; if the pathname already navigates to an attribute, the object manager returns the empty string), "Info" (returns "meta-attributes" such as mode, range and unit), and <none> (i.e., an empty string).

In request 802 of Fig. 8, the server "knows" about an object "HyperMedia Adapter NSK", possibly from receiving a URL from browser 130. Line 820 represents a version of the server (e.g., version 1.0). Line 822 is an "OrmGet" request for object "HyperMedia Adapter NSK". Server 140 requests information from object manager 142 about entity types (Info), Component, and Object (lines 824).

Response 804 is generated by object manager 142 and sent to server 140. The object has four components, no info, and no objects at the same level. As seen in step 312 of Fig. 3, server 140 generates HTML 604 of Fig. 7(c) in accordance with response 804 and sends the generated HTML to browser 130.

Assuming that the user wants to browse information about the Configuration component of object "HyperMedia Adapter NSK", browser 130 sends a request to server 140 to this effect. Server 140 then sends request 806 to the object manager, which responds with response 808. Request 806 is similar to request 802, but the pathname in line 830 is "HyperMedia Adapter NSK/Configuration".

Response 808 includes attributes for the "Info" entity. Thus, the response includes an attribute value, mode, field, and range for each of ten attributes of the component "Configuration". As seen in step 332 of Fig. 3, server 140 generates the HTML of Figs. 6(a) and 6(b) in accordance with response 808 and sends the generated HTML to browser 130 (see Fig. 5).

Assuming that the user wants to change one or more attributes of the Configuration component of object "HyperMedia Adapter NSK", browser 130 sends a request to server 140 to this effect (assuming that the browser is in "set" mode). Server 140 then sends request 810 to the object manager, which responds by sending a status value (not shown).

A format of the OrmSet request is:

OrmSet: pathname
Attribute: name
Value: val

where "name" and "val" respectively, represent an attribute name and an attribute value. This command is shown in line 840. The command can include more than one Attribute/Value pairs.

In the example, request 810 specifies new values for eight attributes of component "Configuration." Assuming that no error occurs when the object manager changes the attribute values, server 140 generates HTML reflecting the new attribute values in accordance with the response and sends the generated HTML to browser 130 (not shown).

A preferred embodiment of the present invention has a server that interfaces with "self describing" (or "introspective") objects. The server sends requests to and receives responses from an ORM (Object Resource Manager). The system may include more than one ORM and more than one server. Each server may "know" about zero or more ORMs. Thus, the system is not centralized and does not necessarily depend on a central point to interface with the objects.

C. The Object Manager

1. Self Describing Objects

Fig. 4 shows an example of object organization in a preferred embodiment of the present invention. Pages of the description, shows examples of an ORM Server Support Library API (Application Program Interface) supported by the object manager to access objects in a preferred embodiment of the present invention. The routines in the API of pages are used by object manager (e.g., ORM 142 of Fig. 1) to receive requests from server 140 and to prepare responses to the requests. It will be understood be persons of ordinary skill in the art that any object manager can be used in conjunction with the present invention, as long as the object manager is capable of communicating with server 140 and of fulfilling GetOrm and SetOrm requests from server 140.

Fig. 10 shows layers of functions available to the object manager. A Protocol layer 1002 handles the ORM protocol, e.g., decodes the request from server 140, initiates the corresponding functions, and assembles an ORM response. Protocol layer 1002 is the lowest layer and drives all calls to the upper layers by calling "registered" functions. A Node layer 1004 handles navigation between nodes, ie.e, parsing the pathname to locate the virtual node, which represents some management entity.

A Handle layer 1006 maps "virtual nodes" to real objects/data. Such a mapping results in a "handle." Handles are explicitly requested and released. An Aspects layer 1008 handles instances that are made up from more than one ORM tree. For example, the "statistics" Component is not a single Component n the tree, but is generated by the object manager. As another example, some attributes depend on others and cannot be modified independently, but have to be treated as a single, atomic operation. These groups of attributes within an instance are called "aspects" and the corresponding Aspect layer is provided to extract and modify groups of attributes within an instance.

An Attribute layer 1010 retrieves or updates a single attribute (of an aspect) and provides the meta information corresponding to this attribute. A Conversion layer handles the actual conversion of attributes between the external (ORM) and the internal (native) presentation. This layer also converts states and bitmaps to "friendly strings."

2. Web Agents

In another preferred embodiment of the present invention, the objects are not self-describing. In such an embodiment, one or more servers 140 in the system performs a "worm" function, i.e., one or more servers 140 follow object-links between objects and save all the information available concerning those objects. When a request is received from browser 130, server 140 sends its collected data to browser 130 (assuming that the collected data is not older than a threshold age value).

In summary, the present invention allows a user to manage objects by way of hypermedia, such as the world wide web. In a preferred embodiment, the objects are self-describing and respond to questions about themselves from one or more object managers. A server communicates with the object manager(s) and generates HTML from responses received from the object manager. Conventional browser software allows a user to indicate which objects he wishes to browse or change. Using a conventional hypermedia request/response protocol, the browser and server communicate to obtain information about objects and their attributes. The server also translates HTML/URLs received from the browser to requests to the object manager. Such a system allows a non-centralized object management system.

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims and equivalents.

5. Browse (readonly) and Change (read write) modes are differentiated by different URL's. For the Change mode an HTML input form is created with user interface elements dependant on the meta information provided with attributes. Dependant from this meta information, simple text input or numeric input fields, popup boxes or radio buttons are generated.
6. A submit of this HTML form results in an HTTP POST request, which appears at the managements adapter with a special URL and the list of name/value pairs from the input form. These values can be checked against the information now retrieved from the object (see above) and a ORM SetAttribute request is send to the respective object. This object initiates the intended state changes or returns an error to the adapter, which then creates a new page, which reflects the outcome of the operation.
7. Access control can be either applied by the generic HTTP adapter, filtering POST methods for example or by the called object itself using principal identifiers in the ORM request.

5.3 Events and Alerts

Compared to SNMP or CMIP, ORM has no event or trap mechanism. With this respect it is much closer to the NSK management protocol called SPI. Instead event and alert support is provided by another mechanism (and object) within MSF, called *Alert Facility*, which is built on top of the *Common Execution Environment* (CEE).

6.0 ORM Protocol

6.1 General Characteristics

The ORM protocol is a simple request response protocol constructed out of lines of ASCII text and terminated by newline, to browse through the managed object and change its state. In this way, it is directly comparable to the HTTP protocol.

It is bytestream oriented, as it contains no length fields or has any fixed structure, but the individual items are separated by special characters, i.e. colons and newlines.

The logical end of a protocol unit is determined by an empty line i.e. a line with a newline character as the first character.

Remark: The decision for a pure ASCII protocol may be surprising, as MSF has a well defined presentation layer/protocol by IDL/PCU/GLU, but ORM is designed to support self contained object management, e.g. all manageable aspects of an object should be described by the managed object (or its manager) itself and just known by it. This implies, that a lot of human readable information has to be shipped with this protocol and using a pure ASCII protocol seemed quite natural. In addition the browsing nature of the protocol would have resulted in a quite unhandy IDL structure of unbounded sequences of any's or unbounded strings, probably with another layer of unbounded types above. Using ASCII strings simplifies the creation of protocol units. Last not least, ORM must also support the entities provided to build the MSF infrastructure itself, e.g. must also be available with and without these execution environments.

Internally the ORM protocol (ORM-P) is a tagged line protocol, as every line starts with a tag word followed by the tag value separated by a colon and terminated by an end of line character (newline). The protocol itself supports boxcarring (e.g. consolidated requests). Errors are reported inline with the data i.e., where they occur, by special error tags.

Request and response units are constructed the same way and the response is merely a "filled out" or "completed" request and as such, the information in it is self describing (i.e. need not necessarily correlated with the request). This is not true for authentication information, which is not mirrored in the response.

The ORM-P basically supports two kinds of operations:

1. get entities (OrmGet)
2. set attributes (OrmSet).

The two keywords OrmGet and OrmSet at the beginning of every sub-request describe the intended operation (i.e. are the operation tags).

In addition to these two basic operations there is a preamble required for every (consolidated) ORM packet, identifying the originators protocol version (tag OrmVersion) and in the case of a request, there is an optional tag, to pass authentication information from the client to the server (OrmWho).

6.2 Pathnames:

Pathnames provide the necessary navigation and identification information to locate a specific aspect of an object (or its internal entities). The different parts of the pathname usually are also the names of the entities in the generated user interface and as such, should be "friendly", descriptive names.

A pathname consists of a "/" separated list of components, which in turn may contain any printable character including whitespace.

For the current version, characters in the pathname components are restricted to the printable set of ASCII characters but may be extended to cover all printable ISO-8859-1 characters in the future (ORM-P is "8-bit clean", but this restriction is just to be more flexible in the choice of user interface construction).

There are two special component names, which follow the POSIX convention for filesystem tree navigation: "." and "..". These are only allowed in conjunction with an OrmPath tag and refer to the current node and to the parent of the current node respectively, where the "current node" must have been defined by a previous ORM sub-request in the same protocol unit (i.e. OrmGet or OrmSet). An attempt to traverse the parent of the root of the tree is treated as an erroneous pathname (compare with "cd" in a POSIX system).

6.3 Error Reporting (OrmError):

Errors are reported, where they are detected, i.e. the error tag (OrmError) usually appears directly behind the error causing tag in the response. This allows some kind of identification of the failed subrequest or protocol item even for large consolidated requests.

An error tag in the response also indicates, where the request was aborted. Previous subrequests in the same ORM packet may have been processed[1]. An error tag (OrmError) has a constant and a variable part, where the constant part identifies the kind of error, the variable part is used for additional hints, what caused the error. As the error is reported with a context, this context provides valuable information for error explanations.

The Format of an error protocol item:

```
OrmError:<decimal errorcode> <stringcode>:<variable part"\n"
```

6.4 Version Support: OrmVersion:

The first line in every protocol unit must identify the highest protocol version number, this protocol unit complies to. The actual version is 1.0!

[1] When the ORM requests are issued via the IDL interface, the error also appears as an exception with the exception detail showing the actual ORM error code.

e.g.

```
        OrmVersion:<major>.<minor>
```

A server/application may respond to this request with an error VersionMissmatch followed by its desired version id.

6.4.1 Protocol Conformance & Protocol Errors:

Unknown tags should be ignored unless they appear at a position, where another tag is required (an attribute tag must be followed by a value tag for example). If the latter occurs, the unidentified or unexpected tag is placed in the response message followed by an error tag with "ProtocolFailure"

6.5 Principal Identification Information: OrmWho

For the purpose of propagating the identification of a principal causing the request or to be used with the request, ORM-P supports a protocol item to ship any kind of (encoded) principal identifier with a request in the following way:

```
        OrmWho:<principal identifier>
```

Up to now, neither the <id-scheme> nor the encoding or interpretation of the "encoded-principal-identifier" are specified in any detail by ORM-P but are up to the application and require an agreement between the ORM-P client and server. This may be subject of change!

6.6 Browse Operation: OrmGet

The operation tag "OrmGet" has to be followed by a colon (":" - the tag separator) and usually is followed by a "pathname". As every ORM protocol item, the operation tag followed by the optional pathname has to be terminated by a newline character. This first line is followed by a list of entity type specifyers requested.

6.6.1 Entity Types

The *OrmGet* request is followed by a list of type specifyers to describe the kinds of entities requested for browsing. Allowed types (and entity specifyers) are:

- *Object*

  requests a list of all known object links at this level. A friendly name and the link-address (NOR) is returned per object.
- *Component*

  A list of all components below the level of the path specified in the GET line is requested. A list of names is returned.
- *Attribute*

A list of attributes for the current node is requested. For every attribute its name (*Name:<name>*) and its stringified value (*Value:*<value) is returned. If the path in the OrmGet line already navigates to a single attribute, the returned name is the empty string.

- *Info* (implies Attributes)

  This type addresses the same type of entities as *"Attributes"*, but here meta information in addition to the name/value pair is requested. as there is Field for identifying the type of input expected, *Mode* to describe the access mode (read, read-write or write) and the so called hints (*Range* and *Unit*), which can be used by the user-interface to generate a more sophisticated presentation of the attribute. These *info* fields are described below in detail in the response section.

- <Empty>

  An empty type specifyer indicates that the validity of the pathname should be checked, but no information is requested.

As all ORM protocol items, every single type specifyer is terminated by a newline.

Examples:

```
1)OrmGet:/telnet/windows\n
  Component\n
  \n

2)OrmGet:/telnet/windows/#pty1/status\n
  Attribute\n
  OrmPath:../../pty2/status\n
  Attribute\n
  \n
3)OrmGet:/telnet\n
  Object\n
  Component\n
  Info\n
  \n
```

Note: Leading blanks in front of the tag or the tag value are ignored as well as lines, whose first character is a "#".

The OrmGet request may be called without a pathname specification (e.g. "OrmGet:\n") in which case the root object itself is referenced and the only valid type specifyer is "Object", which will return the "friendly" name of this object manger and its link address (Note: this link address may be another one, as the request was sent to, i.e. this allows redirecting the management requests to another object reference).

6.6.2 OrmGet Response

The response to a *OrmGet* request merely mirrors the request, but here the type specifyers are used as tags (to type the following entity) followed by the name of the entity (separated by ":") followed by a newline character. The name item is followed by a type dependent list of additional tagged items, describing further properties of this entity.

Although the partial responses below are listed per entity type, they appear in the same response unit in the same sequence as in the corresponding request unit, i.e. if the latter requested entity types by the sequence "Object\nComponent\Attribute\n", then the response will first list all available objects at this level followed by all available components followed by all available attribute/value pairs. If a requested type is not available at the specified level, an empty tag of that type (with a colon) is returned.

The response unit for an *OrmGet* request starts with the common response header (i.e. *OrmVersion:<version>*) followed by the "OrmGet: <pathname>" line followed by any number of the following constructs.

6.6.3 Object Entities:

```
Syntax:
<object-item> ::= 'Object' [':' <string> '\n'
                  <object-link>] '\n'
<object-link> ::= 'Link' ':' <string>

Example:

Object:Network Service Layer\n
Link:<obj-reference>\n
Object:Media Access Layer\n
Link:<obj-reference>\n
\n (if this is the end of the response!)
```

6.6.4 Component Entities: (have no additional properties)

```
Syntax:
<component-item> ::= 'Component' [':' <string>] '\n'
```

Example:

```
Component:Configuration\n
Component:Statistics\n
\n (if this is the end of the response!)
```

6.6.5 Attribute Entities (simple request)

```
Syntax:

<attribute-item> ::= 'Attribute' [':' <string> '\n'
                         <attribute-value>] '\n'
<attribute-value> ::= 'Value' ':' <string>

Example:
      Attribute:Packets sent\n
      Value:1234\n
      Attribute:Packets received\n
      Value:4321\n
```

6.6.6 Attribute Entities (Info request):

```
Syntax:

<attribute-item> ::= 'Attribute' [':' <string> '\n'
                        <attribute-info> | '\n']
<attribute-info> ::= <attribute-value> '\n' (cnt)
                     <attribute-mode>'\n' (cnt)
                     <attribute-field> '\n' (cnt)
                    [<attribute-range> '\n']
                    [<attribute-unit> '\n']
<attribute-mode> ::= 'Mode:' ['RO'|'WO'|'RW'] ['P']
<attribute-field>::= 'Field:' <ORM-fieldtype>
<attribute-range>::= 'Range:' <ORM-range-definition>
<attribute-unit> ::= 'Unit:' <string>

Example:
Attribute:Status\n
Value:Running\n
Mode:RO\n
Field:String\n
Attribute:New Status\n
Value:Running\n
Mode:WO\n
Type:Enum\n
Range:Stopped,Aborted\n
....
```

Here the sequence of the different *info* tags is not relevant, but "Attribute" always starts a new property set for the next attribute. The "Range" and "Unit" tags are optional.

For "ORM-field type" and "ORM-range-definition" see below.

Errors & Exceptions specific to this request:

- NoSuchNode:

  The path specified does not point to a legal virtual node.This error is only reported immediately after a "Orm[Get/Set/Path]:<pathname>" command.

- InvalidOperation:

  The path specifies a node, which can not support the requested entity type, an "Attribute" request for an "Object" node or vice versa for example. This error is reported after a type specifyer, listing the type specifyer (without a colon) followed by a newline followed by the error tag.

6.7 Modification Requests: OrmSet

The set request starts with the set-tag "OrmSet:" followed by a pathname followed by a newline. The pathname at minimum must contain the name of the root e.g. "/<root-name>" (i.e. an empty pathname is not allowed!).

This line is followed by a sequence of line pairs containing the name of the attribute and its value, e.g.

```
Syntax:

<attribute-item> ::= "Attribute" [":" <string> "\n"
                          <attribute-value>] "\n"
<attribute-value> ::= "Value:" <string>

Example:
Attribute:New Status\n
Value:Suspended\n
Attribute:Reset Statistics\n
Value:Yes\n
\n (if this is the end of the protocol unit)
```

6.7.1 OrmSet Responses:

If no error occurred, the response to the *OrmSet* request is a copy of the request itself. Otherwise an error-tag may appear somewhere in the response, and if the underlying request/response protocol permits, the response is flagged with an error indicator.

The effect of an erroneous OrmSet sub-request is application dependant, but it is recommended, that a OrmSet sub-request either succeeds completely or has no effects at all (atomicity).

Any OrmSet requests preceding a failed one are not affected, any subsequent requests are ignored.

6.7.2 Error-Returns:

- NoSuchNode:

  The path specified does not point to a legal virtual node. This error is only reported immediately after a "SET:<pathname>" command.

- NoSuchAttribute:

  The string following an "Attribute:" tag does not identify a legal attribute. The error-tag follows the "Attribute:" tag (including the string).

- ValueOutOfRange

  The value specified is not within the range of legal new values for this attribute. The error tag follows the value-tag line.

- ValueInconsistent

  The set of attribute values in the request were no consistent or contradictory

- InvalidOperation:

  The designated Attribute is not writeable.

- NoPermission:

  The access rights of the requester do not allow to set the designated attribute.(This error and the previous one may have some overlap)

6.8 Request Type Independent Errors:

- ProtocolError:

  If pairs of tagged lines are expected and the sequence of pairs is not completed or an unknown or context illegal tag is detected, this error is generated, following the erroneous tagged line.

- InternalError:

  An internal error in the application/server was the cause, that this request could not be completed. (allocation failure, mangled structures). This indicates a severe error at the server side.

- BufferTooSmall:

  The response buffer specified is too small to return the full response.

- NoSpace:

  Some internal buffer could not be allocated or was to small for the requested operation.

6.9 ORM Attribute Info Descriptions:

Within the ORM protocol there are two ways to retrieve an attribute: the short form returns just the name of the attribute and its value and the long form, returning additional meta information for every attribute, which can be used to create reasonable user interface elements by the ORM client.

The following fields appear n the extended description:

- Field: identifies the kind of field, this attribute should be presented in
- Mode: identifies applicable operations (readonly, read/write, writeonly)
- Range: Provides hints for input checking and for user interface generation
- Unit: a free form field often describing the metric of the value or scale.

6.9.1 ORM Field Types:

Although in principal, the ORM field type item (*Field*) allows any principal character string, the ORM support library and the user interface generator (HTML synthesizer) will only support a limited set of predefined field types, to ease the presentation of attributes. If a field type is not recognized, the default "String" is assumed.

6.9.1.1 Field: Integer

The ORM protocol does not distinguish between unsigned and signed integers, e.g. every ascii string representing an integer may be prefixed by a "-" or a "+". There is also no size information in the field type. Any range restrictions have to be specified in the *Range* section.

Syntax:

Field:Integer

6.9.1.2 Field: Real

The field type *Real* identifies decimal floating point values. The allowed input formats are those of the POSIX 1003.2 scanf function for float and double values.

Syntax:

Field:Real

6.9.1.3 Field:HexOctet

This field type is used to display and enter binary data as pairs of hexadecimal character

Syntax:

Field:HexOctet (a sequence of hexadecimal digits)

6.9.1.4 Compound-Field Types:

The last set of field types allow much finer control of the input, an end user may provide to the ORM-P client side (or the client of the client...). These types are named *Enum* and *Set*, where *Enum* specifies a *"one out of m"* field and *Set* specifies a *"n out of m"* field.

Both types are only valid with an appropriate *Range* field in the hints section, where the possible alternatives must appear in a comma separated list.

These two types often transformed into "Pop-Up" menus (*Enum*) or option lists (*Set*) or similar by the user interface generator.

```
Syntax:

Field:Enum (single selection from "Range:")
Field:Set  (multiple choices from "Range:")
```

6.9.2 ORM Attribute Modes:

To generate reasonable user interfaces (as far as possible without object/component specific knowledge), the generator must know, whether an attribute is "read-only", "read-write" or "write-only". The latter is used to signal to the user interface, that this attribute should be only shown in "change-attribute" frames, if those are distinguished from pure browsing frames. An extension to these basic modes is provided for writable attributes to indicate, that an attribute value is mandatory, by appending the letter "M"

The different modes are simply encoded as two-letter strings followed by an optional "P", e.g.

```
Syntax

- RO      Read-Only
- WO[M]   Write-Only[non-null value mandatory]
- RW[M]   Read-Write[non-null value mandatory]
```

6.9.3 Range Identifiers:

The range identifier, tagged with "Range:" is used as a kind of hint (and therefore it is optional except for the compound fields) to the user interface generator, what kind of input/output field it should generate. In addition the information can be used to check any optional input and give the end-user appropriate responses or hints, if these input checks fail.

The range hints are type specific and as such different conventions are defined to specify valid ranges for an input field. The type independent convention is to separate alternatives by a comma "," and sequences by three subsequent dots "...".:

6.9.3.1 Range Specifications for Integer Fields:

Valid range specifications for the integer types are:

```
Range:1,4,8,16     valid: 1 of the values listed
Range:20...60      valid: all numbers between 20 and 60 including
Range:0...         valid: every integer including 0 (up to typemax)
Range:-20...+20    valid: every integer between -20 and +20 incl.
Range:             valid: every pos/neg integer within type
```

6.9.3.2 Range Specifications fro Floating Point Fields:

Valid range specifications for the real types are:

```
Range:0.1,0.5,0.8        valid: one of the values listed
Range:0.1e-3...0.1       valid: reals between 0.0001 and 0.1
```

6.9.3.3 Range Specifications for String Fields

If the first range value starts with a digit, the range indicates either the maximum or the range of valid string lengths. If the first character is non numeric, the range is interpreted similar to the compound *Enum* field below, i.e. one of these strings may be selected, but a different user interface element may be used (a list box). If the first character of the range string is a comma ",", this provided strings in the comma separated list are treated as examples, where the possible input is not restricted to the given alternatives. A major use of this kind of string selection is in file selection boxes.

```
Syntax:

Range:1...20             valid: strings with minimum length of 1
                         and max length of 20 characters
Range:10                 valid: a string with at max 20 characters.

Range:,file1.c,file2.c   file1.c or file2.c are valid options,but
                         other input is also valid
```

6.9.3.4 Range Specification for Compound Fields:

For the *Enum* and the *Set* type fields lists of alternatives are required in the range section. The comma separated list identifies the different options a user is allowed to select.

```
Syntax:

Range:<comma separated list of alternatives>

Example:

For Enum (choose one of)

Range:STOP,SUSPEND,ABORT

For field type Set (choose n or none of)

      Range:Trace IP,Trace UDP,Trace TCP
```

6.9.4 The Unit Specification:

The "Unit" specification is a free form string and currently not interpreted by the user interface generator. If present, it will append this string behind the value field as one would do with a unit description like "1.4 inches". Another important purpose of this field is for the use with customized object specific management pages (if used within an HTML environment). Here the unit could be used to identify an application specific type for example.

6.10 Navigation Request: Path

This request extends the previous operation (*OrmGet* or *OrmSet*) to a new subtree and follows these tags in its syntax. It may appear everywhere, where a *OrmGet* or *OrmSet* tag may appear, except that it must be preceded by one of these items in the same protocol unit. It usually is only found in sequences of ORM statements resulting from a "Dump" request!

Syntax:

```
OrmPath:<pathname>
```

Semantics: Extends the previous *OrmGet* or a *OrmSet* request into another subtree within the same object.

6.11 Summary of ORM Error Codes:

NoPermission: 1

The current authentication can not be used to perform the requested operation

NoSuchNode: 2

The pathname specified in a Orm[Get/Set/Path] request does not point to a known node.

NoSuchAttribute: 3

The attribute specified in a OrmSet request could not be found

NoSuchObject: 4

The Object specified could not be found.

InvalidOperation: 5

The operation requested is not valid for this type of entity. (Example: attribute is not writable, request components of an attribute).

ProtocolFailure: 6

A sequence of tags was encountered, which could not be parsed and decoded.

VersionMissmatch: 7

The object could not deal with the version of the request packet.

CommunicationError: 8

This is a client side error to map lower level communication errors too, if necessary.

ValueOutOfRange: 9

The value passed in with a set request for an attribute is not within the allowed range and could not be accepted.

ValueInconsistent: 10

The combination of values passed with a set request is not acceptable.

NoSpace: 12

The request could not be completed because of internal space restrictions in the object.

BufferTooSmall: 13

The response to the request exceeds the size of the response buffer provided by the underlying protocol.

InternalError: 14

ApplicationError: 15

These two errors are used to report back implementation problems like corrupt data structures, where the InternalError usually is generated by the ORM support library, the ApplicationError instead is issued by the higher "application" layers.

3.1 ORM Protocol Layer And Upcall Interfaces

This section was generated from <*stdin*> by CDOC on Sun Jan 29 17:00:50 1995.

ORM Application Context

Application Server Capsules may serve different kind of requests and therefor may have multiple domains of objects to be managed listening on multiple ports. Following the ORM model, this may result in multiple parallel independant trees.

The ORM parser supports this by maintaining an application context, which has to be passed to the protocol layer to handle a request (there is also an opaque *call-context*, which may be passed to the protocol layer, but this isn't interpreted by the ssl).

The application context contains beside (an opaque pointer) to the (virtual) root of the virtual tree, mainly a list of tree/application specific function pointers. Before the first request can be passed on to the ORM protocol layer, this context has to be established with the ORM SSL via a call to ORM_ContextInitialize.

Accordingly there exists a function to inform ORM that this application context is not needed anymore (release).

The following lists the function prototype definitions for actual functions to be provided, when establishing a context.

Note: Some functions are defined to return pointers to character strings (ORM_String). If the ORM protocol handler is used it is guaranteed, that the same function will not be called, before the string is copied or otherwise not needed anymore. This allows the use of a single private string buffers per function, if necessary.

3.1.1 Authentication

The following list of functions are included to enable an application to maintain its own authenticated context. The ORM protocol just allows to forward some authentication related information from the client to the server (WHO...). This is passed on to the application layer as is, if encountered by the parser. The actual meaning of this data is application and user interface dependant.

3.1.2 Function Type ORM_AuthenticateFunc

Performs any necessary authentication or preparation of authentication structures. Usually, the authentication information is used to setup some context in the call-context, which is passed to the node/handle layer upcalls. It is up to the application layer to free/clear such context after return from the protocol layer.

Declaration:

```
typedef ORM_Status (*ORM_AuthenticateFunc) (
                    ORM_AppCallContextDef    callcontext,/* in*/
                    ORM_String               authstring  /* in */
```

Fields:

callcontext An opaque pointer to any kind of context, the caller has established.This passed to the node and handle layer.

authstring The string, the client passed in his request, if any. Usually uid:passwd

status ORM_ENoError: if successfull, ORM_EPermissionDenied, if authentication unknown.

3.1.3 ORM_AuthFuncDef

This structure is used to pass the Authentication function to ORM_ContextInitialize

Declaration:

```
typedef struct ORM_AuthFuncTag {
     ORM_AuthenticateFunc          auth;
   } ORM_AuthFuncDef;
```

3.1.4 Virtual Node & Tree Function Types

The following list of functions (function types) are used to access the virtual tree of components, attributes and linked objects. They usually don't deal with application specific data.

3.1.5 Function Type ORM_NodeLookUpFunc

This is the central function for the traversal of *the* tree .

Returns an opaque pointer to a virtual node, which may subsequently be called to retrieve properties or children of specific types.

Declaration:

```
typedef ORM_Status (*ORM_NodeLookUpFunc ) (
         ORM_AppCallContextDef    callcontext,/* in */
         ORM_AppNodeDef           root,       /* in */
         ORM_String               pathname,   /* in */
         ORM_AppNodeDef           *node,      /* out */
         ORM_NodeTypeDef          *nd_type    /* out */
     );
```

Fields:

callcontext is an opaque pointer to the application specific call context provided with the Do_Request function.

root Opaque Pointer to root of virtual tree. This may be NULL, and is taken from the application context.

pathname is a / separated list of component names optionally preceded by the name of the object (e.g. if the first component matches the roots object name, strip it, else take the first component to be a child under the applications root). Support for ***un*x*** style directory navigation . and .. is highly recommended/required. A pathname of .. applied to the root with request type *Object* should return the root name and the actual servers link address (NOR)

node The opaque node pointer, if found

nd_type The ORM_NodeType of the node found

return ORM_ENoError in case of success, or any other ORM error in case of failure.

3.1.6 Function Type ORM_NodeChildNextFunc

Used to subsequently scan the children of a single parent. Returns the next child of type *type* of parent *parent*, which logically follows the child returned by the previous call to *NodeChildNext()*, now passed in as *lastchild*. E.g. If *lastchild* is set to *NULL* the logically first child of this parent is requested. If there are no children (of the requested type), then NULL must be returned with ORM_Status set to ORM_NoError.

Declaration:

```
typedef ORM_Status (*ORM_NodeChildNextFunc ) (
    ORM_AppCallContextDef   callcontext,/* in */
    ORM_AppNodeDef          parent,     /* in */
    ORM_AppNodeDef          lastchild,  /* in */
    ORM_NodeTypeDef         type,       /* in */
    ORM_AppNodeDef          *child,     /* out */
    ORM_String              *name       /* out */
    );
```

Fields:

callcontext is an opaque pointer to the application specific call context provided with the Do_Request function.

parent Opaque pointer to the virtual parent node.

lastchild Opaque pointer to the last child returned by a call to this function (in this request), or NULL to request the first child.

type The type of entity, which is requested (ORM_ObjectType, ORM_ComponentType, ORM_AttributeType or ORM_AnyType).

node	Pointer where to store the reference to the node found
name	Pointer to name of node found.
returns	status value. Possible status values, see below!

3.1.7 Function Type ORM_NodeChildByNameFunc

The little sister of ORM_NodeLookUp. Looks for a child with name *childname* directly under the given parent *parent*. This function is primarily used within the processing of Set- Attribute requests. If there is no child with this name, return NULL and an error status (see below)

Declaration:

```
typedef ORM_Status (*ORM_NodeChildByNameFunc ) (
          ORM_AppCallContextDef     callcontext, /* in */
          ORM_AppNodeDef            parent,      /* in */
          ORM_String                childname,   /* in */
          ORM_AppNodeDef            *child,      /* out */
          ORM_NodeTypeDef           *child_type  /* out */
```

Fields:

parent	Opaque pointer to the virtual parent node.
childname	Name of the child (attribute), i.e. every printable char except '/'
child	Pointer where to store the reference to the node found
child_type	Pointer to type of node found.
returns	ORM_ENoError if child was found, else ORM_ENoSuchNode.

3.1.8 Function Type ORM_NodeTypeGetFunc

returns the type (enum ORM_NodeTypeDef) of the given node.

Declaration:

```
typedef ORM_NodeTypeDef (*ORM_NodeTypeGetFunc ) (
    ORM_AppNodeDef          node         /* in */
    );
```

Fields:

node	a pointer to a virtual node
returns	a valid type or ORM_NodeTypeUnknown.

3.1.9 Function Type ORM_NodeNameGetFunc

returns the name (ORM_String) of the given node.

Declaration:

```
typedef ORM_String (*ORM_NodeNameGetFunc ) (
    ORM_AppNodeDef        node        /* in */
    );
```

Fields:

node	a pointer to a virtual node
returns	a valid null terminated string of characters or NULL

3.1.10 Function Type ORM_NodeNotFoundTrapFunc

This function is kind of special by providing the application layer a chance, if the lookup of a node failed,to create that node.

Normally, referencing a non-existent node in the pathname of an ORM request is treated as an error, except this is an internal ORM restore request. Reloading an ORM tree into an application may encounter subtrees, which where dynamically created by the application during a previous run (usually via a *New* subtree).

This function is totally application dependant and is not covered by the ORM-SSL other than via this hook.

Declaration:

```
typedef ORM_Status (*ORM_NodeNotFoundTrapFunc) (
    ORM_AppNodeDef        parent,     /* in */
    ORM_String            name,       /* in */
    ORM_RequestTypeDef    request,    /* in */
    ORM_AppNodeDef        *newnode    /* out */
    );
```

Fields:

parent	Reference to parent node
name	Name of node not found under this parent.
request	Kind of ORM request (get/set/dump/restore) causing this lookup failure.
newnode	Where to store the reference to the new node, if one was created.
returns	ORM_ENoError, if a node with the given name was created else ORM_ENoSuchNode.

3.1.11 Structure ORM_NodeFuncDef

This structure bundles the virtual tree related functions for passing to ContextInitialize

Note: The ORM_NodeNotFoundTrapFunc is not included in this function array, because it is application special anyway and must be passed explicitly, see ContextInitialise()

Declaration:

```
typedef struct ORM_NodeFuncTag {
        ORM_NodeLookUpFunc          lookup;
        ORM_NodeChildNextFunc       childnext;
        ORM_NodeChildByNameFunc     childbyname;
        ORM_NodeTypeGetFunc         typeget;
        ORM_NodeNameGetFunc         nameget;
        } ORM_NodeFuncDef;
```

3.1.12 Application Handles

The following two function types are used to link the virtual nodes in the tree to (parts of) actual application data instances, visible to the ORM support layer as opaque handles. When an application handle is requested from the application layer, *real things* happen to start and it is assumed, that the instances are valid and available, until explicitly released by the ORM layer. The handles together with the aspect (identifying the type of handle to the application) will be passed to the application specific functions, when actual values have to be accessed (either for *get* or *set*). If these functions are not set in the ORM context, NULL will be passed into those calls for both, the handle and the handleclass.

3.1.13 Function Type ORM_HandleGetFunc

Request (and lock) an actual handle (pointer to an application level instance) and a handleclass based on the current virtual node and the current principal.

Declaration:

```
typedef ORM_Status (*ORM_HandleGetFunc) (
        ORM_AppCallContextDef   callcontext,/* in */
        ORM_AppNodeDef          node,       /* in */
        ORM_RequestTypeDef      op,         /* in */
        ORM_AppHandleDef        *handle,    /* out */
        ORM_AppAspectDef        *aspect     /* out */
        );
```

Fields:

callcontext is an opaque pointer to the application specific call context provided with the Do_Request function.

node Pointer to current Node.

op Operation Code, e.g. ORM_Request

handle Pointer, where to store the handle reference *aspect* Pointer, where to store the aspect reference

returns ORM_ENoError if no error occured or any of the ORM error codes.

3.1.14 Function Type ORM_HandleReleaseFunc

Returns a given handle back to the application layer. This should be more understood as an *unlock* operation than a free!

Declaration:

```
typedef void (*ORM_HandleReleaseFunc ) (
    ORM_AppCallContextDef   callcontext,/* in */
    ORM_AppHandleDef        handle,     /* in */
    ORM_AppAspectDef        aspect,     /* in */
    ORM_RequestTypeDef      op          /* in */
    );
```

Fields.

callcontext is an opaque pointer to the application specific call context provided with the Do_Request function.

handle a handle obtained via a call to HandleGet

aspect Aspect as returned from HandleGet

op Operation Code, e.g. ORM_Request

3.1.15 Function Type ORM_ObjectLinkGetFunc

Retrieve the Object Link from a node of type Object given the node, the handle and the aspect. The standard Handle Layer functions just return the link stored in the corresponding field in the node struct.

Declaration:

```
typedef ORM_Status (*ORM_HandleObjectLinkGetFunc ) (
    ORM_AppNodeDef       node,      /* in */
    ORM_AppHandleDef     handle,    /* in */
    ORM_AppAspectDef     aspect,    /* in */
    [illegible]          *link      /* out */
    );
```

Fields:

node Reference to node of Object Type

handle Reference to application defined handle as returned from HandleGet

aspect Reference to application defined aspect as returned from Han-

link Location where to store the reference to the stringified link information

returns ORM_ENoError if successfull, else ORM_InvalidOperation, if the node is not of type Object

3.1.16 Function Type ORM_AttributeDescrGetFunc

Retrieve the opaque reference unique to a node of type Attribute (usually the attribute descriptor), given the node, the handle and the aspect. The standard Handle Layer functions just return the pointer stored in the corresponding field in the node struct.

Declaration:

```
typedef ORM_Status (*ORM_HandleAttributeDescrGetFunc ) (
    ORM_AttrNodeDef            node,        /* in */
    ORM_AppHandleDef           handle,      /* in */
    ORM_AppAspectDef           aspect,      /* in */
    ORM_AppAttributeDescrDef   *attribdesc  /* out */
    );
```

Fields:

node Reference to node of Object Type

handle Reference to application defined handle as returned from HandleGet

aspect Reference to application defined aspect as returned from HandleGet

attribdesc Location where to store the reference to the attribute information

returns ORM_ENoError if successfull, else ORM_InvalidOperation, if the node is not of type Object

3.1.17 Structure ORM_HandleFuncDef

This structure bundles the handle related functions for passing to ContextInitialize

Declaration:

```
typedef struct ORM_HandleFuncTag {
    ORM_HandleGetFunc                       get;
    ORM_HandleReleaseFunc                   release;
    ORM_HandleObjectLinkGetFunc             link;
    ORM_HandleAttributeDescrGetFunc attrib;
    } ORM_HandleFuncDef;
```

3.1.18 Accessing Application Data: Aspects

the following group of functions (function types) has to be provided to access actual values of the application either for retrieval or for updating. All functions in this group are mandatory, if the ORM protocol layer is used.

3.1.19 Function Type ORM_AspectCallGetFunc

This function retrieves an *aspect* from the application layer, e.g. a reference to a blob of native application data (a pointer to a (part of) an application data structure, or a response buffer ....). The ORM protocol layer calls this function once for every unique handle/aspect combination (and not per Attribute) within a single AttributeGet Request._ If the HandleGet Function returns a different pair or there are no more attribute nodes to process, the current aspect is released!

Declaration:

```
typedef ORM_Status (* ORM_AspectCallGetFunc) (
    ORM_AppHandleDef      handle,      /* in */
    ORM_AppAspectDef      aspect,      /* in */
    ORM_AppDataPtrDef     *current     /* out */
    );
```

Fields:

handle	Handle as retrieved from HandleGet
aspect	Aspect Reference, as retrieved from HandleGet
current	Where to store the reference to the current value (opaque)

3.1.20 Function Type ORM_AspectCallInitFunc

This function requests an *aspect* container from the application layer, e.g. a reference to a blob, where new attribute values can be selectivly written to to perform AttributeSet requests. In addition hte application layer may return a reference to the current aspects values (cmp CallGet), which is passed unchanged to the CallSet routine. The ORM protocol layer calls this function once for every unique handle/aspect combination (and not per Attribute) within a single AttributeSet Request. If the HandleGet Function returns a different pair for a node or there are no more attribute nodes to process, the CallSet function is called (Note: AspectRelease is only called for aspects retrieved via CallGet!) The

ORM SSL Implementation of these functions copies the current values and returns a reference to this copy in *new* and a reference to the current values in *current*.

Declaration:

```
typedef ORM_Status (* ORM_AspectCallInitFunc) (
    ORM_AppHandleDef      handle,       /* in */
    ORM_AppAspectDef      aspect,       /* in */
    ORM_AppDataPtrDef     *new,         /* out */
    ORM_AppDataPtrDef     *current      /* out */
    );
```

Fields:

handle Handle as retrieved from HandleGet

aspect Aspect Reference, as retrieved from HandleGet

new Where to store the reference to the native blob to update with new attribute values (opaque)

current Where to store the reference to the current aspect (opaque)

3.1.21 Function Type ORM_AspectCallSetFunc

This function is called to actually apply the new attribute values for the current aspect by the application layer. It is up to the aspect/application layer, to check the values in the request structure for validity and consistency and to determine which attributes got new values (by comparison with the *current* values). In addition it is the responsibility of the aspect/application layer to deallocate any structures allocated by AspectCallInit. Only if the Set-Function is not called, the call to AspectRelease is performed.

The ORM protocol layer calls Set-function once for every unique handle/aspect combination (and not per Attribute) within a single AttributeSet Request. If the HandleGet Function returns a different pair for a node or there are no more attribute nodes to process, the CallSet function is called (Note: AspectRelease is only called for aspects retrieved via CallGet!) The ORM SSL Implementation of these functions copies the current values and returns a reference to this copy in *new* and a reference to the current values in *current*.

Declaration:

```
typedef ORM_Status (* ORM_AspectCallSetFunc) (
    ORM_AppHandleDef      handle,       /* in */
    ORM_AppAspectDef      aspect,       /* in */
    ORM_AppDataPtrDef     new,          /* in */
    ORM_AppDataPtrDef     current,      /* in */
    ORM_String            *rsdetail     /* out */
    );
```

aspect	Aspect Reference, as retrieved from HandleGet
request	Where to store the reference to the native blob to update with new attribute values (opaque)
current	Where to store the reference to the current aspect (opaque)
rsdetail	Where to store a textual hint, why the call failed, if any.
returns	ORM_ENoError if new values could be applied successfully, else ORM_ERange.

3.1.22 Function Type ORM_AspectReleaseFunc

Used to tell the application layer, that the reference retrieved via an AspectGet or AspectInit call is no longer needed anymore by the ORM layer. This function is called, when GetHandle returns a new handle aspect call within a AttributeGet processing or a conversion in an AttributeSet processing failed.

Declaration:

```
typedef void (* ORM_AspectReleaseFunc) (
     ORM_AttrHandleDef      handle,      /* in */
     ORM_AppAttrHdlDef      aspect,      /* in */
     ORM_AppDataPtrDef      current,     /* in */
     ORM_RequestTypeDef     reqtype      /* in */
     );
```

Fields:

handle	Handle as retrieved from HandleGet
aspect	Aspect Reference, as retrieved from HandleGet
current	Reference to data as returned from AspectCallInit or AspectCallGet.
reqtype	ORM_RequestGet or ORM_RequestSet depending whether this dataptr resulted from an AspectGet or AspectInit call.

3.1.23 ORM_AspectFuncDef

This function groups the function pointers of the aspect layer

Declaration:

```
typedef struct ORM_AspectFuncTag {
        ORM_AspectCallGetFunc    callget;
        ORM_AspectCallInitFunc   callinit;
        ORM_AspectCallSetFunc    callset;
        ORM_AspectReleaseFunc    release;
        } ORM_AspectFuncDef;
```

3.1.24 Attribute Functions

The following group of functions is called to actually perform the the single attribute Get/Set and the corresponding conversions between the applications native and the ORM (ascii) presentation.

3.1.25 Data Structure: ORM_AttributeInfoDef

This structure is used to return the all the meta information and the actual value of an attribute. It is passed by reference to the application/attribute layer to be filled. Note: The string pointers do not point to valid buffers, when passed to the attribute layer!

```
<mode>  [illegible] RW]
<type>  INT[1 2 4 8], REAL[32|64], STRING, HEXOCT,
        [illegible], MCHOICE
<value> The current value in its ascii presentation
<range> Optional: The range string
<unit>  Optional: The unit string
```

Declaration:

```
typedef struct ORM_AttributeInfoTag  {
        ORM_String          value;
        ORM_[illegible]     mode;
        [illegible]         field;
        ORM_String          range;
        ORM_String          unit;
        } ORM_AttributeInfoDef;
```

3.1.26 Function Type ORM_AttributeNativeToStringFunc

This function converts the applications native value of an *attribute*, specified by *handle*, *aspect* and the attribute descriptor to a C-string (ORM_String).

Declaration:

```
typedef ORM_Status (*ORM_AttributeNativeToStringFunc)(
        ORM_AppHandleDef           handle,         /* in */
        ORM_AppAspectDef           aspect,         /* in */
        ORM_AppAttribDescrDef      attribdescr,    /* in */
        ORM_AppDataPtrDef          dataptr,        /* in */
        ORM_String                 *strvalue       /* out */
        );
```

Fields:

handle Handle as obtained from the last call to HandleGet or NULL.

aspect Aspect as returned from the last call to HandleGet or NULL

attribdescr *Attribute Descriptor as returned form AttribDescrGet* call.

dataptr Opaque Pointer as returned from AspectGetCall.

strvalue Where to store the reference to the converted value.

returns ORM_ENoError (Null) if conversion was successfull, else a valid ORM Error return code.

3.1.27 Function Type ORM_AttributeNativeToInfo

This function performs the same as the previous function *ORM_AttributeNativeToString, except that it also provides* the additional meta information to this attribute, as far as available.

Declaration:

```
typedef ORM_Status (*ORM_AttributeNativeToInfoFunc)(
        ORM_AppHandleDef           handle,      /* in */
        ORM_AppAspectDef           aspect,      /* in */
        ORM AppAttribDescrDef      attribdescr, /* in */
        ORM_AppDataPtrDef          dataptr,     /* in */
        ORM_AttributeInfoDef       info         /* in, indirect out */
        );
```

Fields:

handle Handle as obtained from the last call to HandleGet or NULL.

aspect Aspect as returned from the last call to HandleGet or NULL

attribdescr *Attribute Descriptor as returned form AttribDescrGet* call.

dataptr Opaque Pointer as returned from AspectGetCall.

extref Pointer to structure, where to store the string references.

returns ORM_ENoError (Null) if conversion was successfull, else a valid ORM Error return code.

3.1.28 Function Type ORM_AttributeStringToNativeFunc

This function converts an ORM_String value for an attribute into the applications native presentation. The conversion should be done into the structure (dataptr) obtained by a call to AspectCallInit().

Declaration:

```
typedef ORM_Status (*ORM_AttributeStringToNativeFunc)(
        ORM_AppHandleDef          handle,      /* in */
        ORM_AppAspectDef          aspect,      /* in */
        ORM_AppAttribDescrDef     attribdescr, /* in */
        ORM_AppDataPtrDef         dataptr,     /* in, indirect out */
        ORM_String                strvalue     /* in */
        );
```

Fields:

handle Handle as obtained from the last call to HandleGet or NULL.

aspect Aspect as returned from the last call to HandleGet or NULL

attribdescr *Attribute Descriptor as returned form AttribDescrGet* call.

dataptr Opaque Pointer as returned from AspectGetCall.

strvalue New value as a C-String (ascii).

returns ORM_ENoError (Null) if conversion was successfull, else a valid ORM Error return code.

3.1.29 Structure ORM_AttributeFuncDef

This structure bundles the attribute related functions for passing to ContextInitialize

Declaration:

```
typedef struct ORM_AttributeFuncTag {
    ORM_AttributeStringToNativeFunc      stringtonative;
    ORM_AttributeNativeToStringFunc      nativetostring;
    ORM_AttributeNativeToInfoFunc    infotostring;
    } ORM_AttributeFuncDef;
```

3.1.30 Structure ORM_ContextDef

This is an internal structure to ORM and opaque to the application layer. It stores the function pointers and the information of the root node.

Note: This structure and the related procedure definitions may change *authfuncs* Pointer to list of authentication related functions or NULL, if no application specific authentication is needed.

notfound No description

3.1.32 ORM_ContextRelease

Release an Application Context.

Prototype:

```
void
ORM_ContextRelease( ORM_ContextDef contxt);
```

Parameters:

contxt Pointer to application context as obtained from ORM_ContextInitialize

3.1.33 ORM_DoRequest

This function calls the protocol layer to parse an ORM request received and act on it accordingly via upcalls to functions in the application context, i.e. this is the function to be dispatched, when ORM requests are received on a server port.

Prototype:

```
ORM_Status
ORM_DoRequest(
    ORM_ContextDef      appctxt,
                        *callctxt,
    ORM_RequestDef      request,
    long                reqlen,
    ORM_ResponseDef     response,
    long                *maxresp
    );
```

Parameters:

appctxt The application context reference as returned from ORM_ContextInitialize.

callctxt An arbitrary call context (reference) maintained by the application layer and passed to the authentication, node and handle upcalls.

request Pointer to received ORM request

reqlen Length of request buffer in bytes

response Pointer to allocated response buffer

maxresp	Reference to maximum response buffer length in bytes, on return, points to number of bytes used in response buffer

3.2 ORM Node Layer

This section was generated from *<stdin>* by CDOC on Fri Jan 27 19:59:34 1995.

The ORM Node layer adds another level of ORM application/server support, as it actually maintains a tree structure to access the application level datastructures.

This level is accessed from the application/server level via the *ORM_Node*... functions to actually build/destroy the tree of objects, components and attributes.

On the other side it is called from the protocol level and frees up the application to provide the appropriate functions for navigation and name space/entity management itself.

3.2.1 Application Handles

The nodes of the node layer provide a tree structured view to application/server level data, but they (usually) do not contain the actual data. A link to the actual instances of application level data is maintained by *handles* and *aspects*. Both are opaque to the ORM-Node level but are interpreted at the layer on top of ORM-Node. Typically the handle is a pointer to some application level instance, and the aspect is a pointer, index or type identifier, which identifies the type of the instance

3.2.2 The ORM_Node Structure

Instances of this structure maintain the tree of virtual components, objects and attributes

Every node has a name and a type, identifying the three different entity types: Object, Component or Attribute. Object and Attribute nodes are leaf nodes, e.g. they can't have children.

In addition, every node has a parent and a next pointer, to link the actual tree structure. Only component nodes have a pointer to the list of children.

Object Nodes have an additional attribute, called the Link (or Link-Info which usually is a stringified NOR).

Attribute Nodes reference a single attribute by, which is characterize by additional information like

- a value type, which describes the kind of value e.g. integer (different sizes), real (sizes!), string, *single-selection* or *multiple choice*
- a value mode, specifying this attribute as read-only read-write, write-only or persistant.
- *hints* section, which contains additional information for use by the user-interface creator, e.g. valid ranges for this value and a unit string. Both values are optional.

The nodes provide a tree structured view to application/ server level data, but they (usually) do not contain the actual data.

3.2.3 Struct NodeDef

Declaration:

```
typedef struct ORM_NodeTag {
    ORM_NodeTypeDef      type;
    short                flag;
    char                 *name;
    struct ORM_NodeTag   *parent;
    struct ORM_NodeTag   *next;
    ORM_AppHandleDef     handle;
    ORM_AppAspectDef     aspect;
    union {
        struct
            struct ORM_NodeTag *first;
            struct ORM_NodeTag *last;
            } comp;
        struct
            void     *descr;
            } attrib;
        struct {
            char     *link;
            } object;
        } u;
    } *ORM_NodeDef;
```

Fields.

type indentifies the type of entity, this node describes, i.e. ORM_NodeType[Object, Component Attribute, Unknown]

flag Internal use

name The name of the node (object, component or attribute name

parent pointer to the parent in the tree, NUL for the root of the tree.

next pointer to next sibbling in chain. This defines the order in which nodes of a given type appear in the response

handle an opaque pointer for use by the upper layers

aspect another opaque identifier for use by the upper layers

u.comp union variant for component nodes

u.comp.first pointer to first child of this component node

u.comp.last pointer to last child of this component node

u.attrib union variant for attribute nodes

u.attrib.descr opaque pointer for use by upper layers *u.object.link* pointer to stringified link-address of this object (NOR), e.g. the *hyperlink*

3.2.4 ORM_NodeCreate

Creates a new unlinked node. Usually only used by convenience functions and to create the root node.

Prototype:

```
ORM_NodeDef
ORM_NodeCreate( ORM_String        name,    /* in */
                ORM_NodeTypeDef   type     /* in */
        );
```

Parameters:

name The name of this node (for navigation)

type The type of this node. This type also determines which functions - can be applied to this node later on.

3.2.5 ORM_NodeDelete

Deletes the given node and all its children e.g. returns the space allocated Note: if the nodes parent pointer is not NULL, the node will not be deleted.

Prototype:

```
int
ORM_NodeDelete( ORM_NodeDef node      /* in */
        );
```

Parameters:

node The node (and the subtree) to delete

3.2.6 ORM_NodeAttach

Attaches a node (and its subtree) into an existing tree as a new subtree. Every node (subtree) is in at most 1 tree!

Prototype:

```
int
ORM_NodeAttach( ORM_RelationDef relation,        /* in */
                ORM_NodeDef     relative,        /* in */
                ORM_NodeDef     subtree          /* in */
                );
```

Parameters:

relation : Flag either ORM_NodeSibbling or ORM_NodeChild, specifying the role of the *relative* node, e.g. its a sibbling or its the parent of the subtree to attach. If its a parent, the new node will be attached at the end of all children, if its a sibbling, it will be placed right before this child.

relative : an existing node, either parent of sibbling

subtree No description

3.2.7 ORM_NodeDetach

Detaches a subtree from the current root tree. This always has to be called, before a subtree is actually deallocated. The subtree may also be reattached in the same tree again after this call

Prototype:

```
void
ORM_NodeDetach(     ORM_NodeDef subtree       /* in */
          );
```

No parameter descriptions are available.

3.2.8 ORM_NodeHandleSet

Sets the handle in the given node (see also ORM_Node<convenience functions>)

Prototype:

```
void
ORM_NodeHandleSet(  ORM_NodeDef          node,        /* in */
                    ORM_AppHandleDef     handle       /* in */
                    );
```

Parameters:

node Reference to node structure of any type.

handle Reference to opaque handle.

3.2.9 ORM_NodeHandleGet

Retrieves the handle from a given node

Prototype:

```
int
ORM_NodeHandleGet(  ORM_NodeDef          node,        /* in */
                    ORM_AppHandleDef     *handle      /* out */
                    );
```

No parameter descriptions are available.

3.2.10 ORM_NodeAspectSet

Sets the aspect in the given node (see also ORM_Node<convenience functions>)

Prototype:

```
void
ORM_NodeAspectSet(  ORM_NodeDef           node,         /* in */
                    ORM_AppAspectDef      aspect        /* in */
                    );
```

Parameters:

node Reference to node structure of any valid node type.

aspect Reference to opaque aspect description.

3.2.11 ORM_NodeAspectGet

Retrieves the aspect from a given node

Prototype:

```
int
ORM_NodeAspectGet(  ORM_NodeDef           node,                 /* in */
                    ORM_AppAspectDef      *aspect               /* out */
                    );
```

No parameter descriptions are available.

3.2.12 ORM_NodeAttributeDescrSet

Sets the attribute description of an attribute node

Prototype:

```
int
ORM_NodeAttributeDescrSet(  ORM_NodeDef               node,       /* in */
                            ORM_AppAttribDescrDef     attrib      /* in */
                            );
```

Parameters:

node Reference to node structure of type Attribute.

attrib Reference to opaque attribute description

3.2.13 ORM_NodeAttributeDescrGet

Gets the attribute description of an attribute node

Prototype:

```
int
ORM_NodeAttributeDescrGet(  ORM_NodeDef              node,       /* in */
                            ORM_AppAttribDescrDef   *attrib      /* out */
                            );
```

No parameter descriptions are available.

3.2.14 ORM_NodeObjectLinkSet

Sets the link of an object node

Prototype:

```
int
ORM_NodeObjectLinkSet(  ORM_NodeDef node,          /* in */
                        ORM_String  link           /* in */
                        );
```

Parameters:

node Reference to node structure of type Object.

link Stringified version of the address/nor to call this object.

3.2.15 ORM_NodeObjectLinkGet

Gets the linkaddress of an object node

Prototype:

```
int
ORM_NodeObjectLinkGet(  ORM_NodeDef node,          /* in */
                        ORM_String  *link          /* out */
                        );
```

No parameter descriptions are available.

3.2.16 ORM_NodeObjectAdd

for an explanations of paramters, see above. Return created node if operation succeeded else NULL.

Prototype:

```
ORM_NodeDef
ORM_NodeObjectAdd(          ORM_RelationDef       relation,        /* in */
                            ORM_NodeDef           relative,        /* in */
                            ORM_String            name,            /* in */
                            ORM_AppHandleDef      handle,          /* in */
                            ORM_AppAspectDef      aspect,          /* in */
                            ORM_String            linkaddr         /* in */
                   );
```

No parameter descriptions are available.

3.2.17 ORM_NodeComponentAdd

Prototype:

```
ORM_NodeDef
ORM_NodeComponentAdd(       ORM_RelationDef       relation,        /* in */
                            ORM_NodeDef           relative,        /* in */
                            ORM_String            name,            /* in */
                            ORM_AppHandleDef      handle,          /* in */
                            ORM_AppAspectDef      aspect           /* in */
                            );
```

No parameter descriptions are available.

3.2.18 ORM_NodeAttributeAdd

Prototype:

```
ORM_NodeDef
ORM_NodeAttributeAdd(       ORM_RelationDef       relation,        /* in */
                            ORM_NodeDef           relative,        /* in */
                            ORM_String            name,            /* in */
                            ORM_AppHandleDef      handle,          /* in */
                            ORM_AppAspectDef      aspect,          /* in */
                            ORM_AppAttribDescrDef attribdescr      /* in */
                            );
```

No parameter descriptions are available.

3.3 ORM Aspect Layer

This section was generated from *<stdin>* by CDOC on Sun Jan 29 17:00:51 1995.

The ORM aspect layer adds another level of ORM application/server support on top of the ORM Node/Handle layer, and supports the retrieval and modification of aspects, i.e. groups of attributes from or into application data structures, once those have been registered with this layer.

This level has no additional (down-call) functions but defines data structures to be provided by the application layer. These are then accessed/used by the aspect upcall functions, if those have been registered with the ORM protocol layer.

The Aspect layer implementation of the ORM-SSL works as follows:

On AspectCallGet requests, just a pointer is returned which points at offset bytes (as set in the aspect descriptor) from the beginning of the handle. On AspectCallInit calls, a copy of the aspect, e.g. size bytes from the area pointed to by handle, starting from offset, is taken into a private memory area. This copy is then passed to the Attribute conversion routines to write the new values into. On AspectCallSet calls, the application level set function as denoted by the aspect descriptor is called and the private copy (request structure) is released afterward.

3.3.1 Function Type ORM_AspectSetFunc

This function is called from the aspect layer to actually apply the new attribute values to the application layer and/or initiate the requested state changes. This function usually should not block, e.g. should not wait until the initiated state change is completed. Any kind of intermediate state should instead be visible to a client on request (i.e. not STOPPED -> STARTED, but STOPPED -> STARTING -> STARTED, if starting implies a *heavier* operation.

Declaration:

```
ORM_Status
typedef (*ORM_AspectSetFunc)(
     ORM_AspHandleDef             handle,         /* in */
     ORM_AspectDescrListDef       aspect,         /* in */
     ORM_AppDataPtrDef            request,        /* in */
     ORM_AppDataPtrDef            current,        /* in */
     ORM_String                   *errortext      /* out */
     );
```

Fields:

handle the handle as returned from HandleGet

aspect Reference to the aspectdescr.

request Copy of the aspect as described by the aspectdescr updated with new values.

current Reference to aspect within handle

errortext Where to store a pointer to a short textual description if the requested values could NOT be applied.

returns ORM_ENoError if all new values could be applied, or ORM_EParameterList if paremter set is inconsistent or ORM_EMissingAttribute if a mandatory attribute is NULL.

3.3.2 The ORM_AspectDescrDef

This descriptor maintains information about the application data structure (usually references by the ORM_AppHandle) or parts of it. It describes the binary size, the offset within the handle, and contains pointers to functions to actually retrieve or modify this aspect of the application instance.

Note: It is currently open, whether there should be a procedural interface to set up the aspect descriptor instead of providing a structure type definition to be passed initialized by the application code.

Declaration:

```
typedef struct ORM_AspectDescrTag {
        char                    *name;
        size_t                  offset;
        size_t                  size;
        long                    flag;
        ORM_AspectSetFunc       setf;
        long                    appid;
        void                    *appext;
        } ORM_AspectDescrDef;
```

Fields:

name Pointer to name string, for identification mainly.

offset The offset in bytes within the instance, where this aspect starts. This usually is the offset of a sub structure in the instance.

size The size in bytes of the instance, the application handle pointer points to. For set-requests, the container for the new value is created by copying the handle, and inserting the new values in it.

flag If set to ORM_AspectGetIndirect, the offset indicates the offset to a pointer, pointing to another structure of the above size.

setf Pointer to function, which is called to apply (a set of) new values to an application instance.

appext any value of pointer size the application wants to store with the aspect. This may be used to store a create_aspect function pointer.

appid Opaque identifier, which may be used by the applications layer

3.4 ORM Attribute Layer

This section was generated from *<stdin>* by CDOC on Fri Jan 27 19:59:34 1995.

The ORM Attribute layer adds another level of ORM application/server support on top of the ORM node layer, by providing (list of) attribute descriptors, which simply initialized by the application code, allowe automatic conversion and generation of the attribute meta information, requested by the ORM protocol layer.

The implementation of the attribute layer in the ORM SSL assumes, that it is converting to and from a binary blob of data, identified by the (lower level) aspect descriptor. The goal of this layer is to reduce the coding effort needed by the application writer at this layer, just to provide some initialized descriptors and pass them to the ORM SSL via single calls per every instance created.

3.4.1 The ORM_AttributeDescriptorDef

This data structure describes a single attribute, e.g. its native type and mode, its size, pointers to conversion functions. In addition it maintains hooks for preset meta-info like - *Unit* and *Range*.

Declaration:

```
typedef struct ORM_AttributeDescrTag {
    ORM_String                          name;
    ORM_AttribTypeDef                   datatype;
    ORM_AttribModeDef                   accessmode;
    ORM_String                          range;
    ORM_String                          unit;
    size_t                              offset;
    size_t                              size;
    ORM_ConverterNativeToStringFunc     nativetostring;
    ORM_ConverterStringToNativeFunc     stringtonative;
    ORM_AppConverterArgDef              convarg;
    } ORM_AttributeDescrDef;
```

Fields:

name — The name of the attribute.

datatype — The type of data of this attribute (ORM_AttributeTypeDef). This is a superset of the data types, the ORM protocol defines and used to determine implicit conversion routines.

mode — The allowed access modes of this attribute out of ORM_AttribMode values, e.g. read-only, write- only, read-write.

range — A string describing the allowed ranges for new values for read-write or write-only attributes only. This is a *ORM hint*, and as such optional

unit — A unit string (usually *ms*, *Mb*, etc.) which may be used by object specific user interface generators in any way, and by default if present is placed behind the attribute value. This is also an *ORM hint* and as such optional.

conversion function A function pointer to an application specific conversion function, to convert between native and ORM presentations. *Note:* This is not to be confused with the similar functions of the ORM_Context structure. For the <conversion-function> to be called, the ORM_Node conversions functions have to be setup in the ORM_Context.

conversion-arg An opaque pointer to any argument, the conversion function may need to convert this attribute.

3.4.2 ORM_AttributeCreate

This function combines several actions required to register an attribute of a (new) instance with the ORM SSL, i.e. it creates an attribute node under the given parent (which must be of ORM_NodeTypeComponent) and attaches the attribute description and the handle information to it.

Prototype:

```
int
ORM_AttributeCreate(
                ORM_NodeDef                 relative,    /* in */
                ORM_RelationDef             relation,    /* in */
                ORM_AttributeDescrListDef   attrib-descr,/* in */
                ORM_AspectDescrListDef      aspectdescr,/* in */
                ORM_AppHandleDef            handle,      /* in */
                ORM_NodeDef                 *new         /* out */
                );
```

Parameters:

relative pointer to relative node. If *relation* is set ORM_NodeIsParent, then this has to be a node of ORM_NodeTypeComponent. If *relation* is set to ORM_NodeIsSibbling, then this node can be of any valid node type.

relation Either ORM_NodeIsParent, if the node *relative* should be the parent of the new attribute node, or ORM_NodeIsSibbling, if the new attribute node should be inserted after the *relative* node as a sibbling.

attribdescr No description

aspectdescr No description

handle Pointer to the application instance this attribute belongs to or ORM_HandleInherit (-1), if the handle should be taken from the parent (or its parent and so on).

new Pointer to new attribute node or NULL on failure.

3.4.3 ORM_AttributeDestroy

This function detaches the attribute node from the tree of nodes if any, deletes the node structure and deletes any depending structures, i.e. the attribute descriptor.

In the current implementation this function maps directly to ORM_NodeDestroy, but nevertheless this function should be called for attribute nodes created with functions of this layer to be able to deallocate any dynamic memory.

Prototype:

```
int
ORM_AttributeDestroy(    ORM_NodeDef attrnode);
```

Parameters:

attrnode Pointer to attrbute node.

3.4.4 ORM_AttributeListCreate

This is another convenience functions to add a list of attributes to a component. The given node must be a of component type and is used as the parent for the new list of attributes (which is appended to the end of the list of child-nodes). The pointer to the attribute descriptor now points to an array of those descriptors, where the end of the array is marked by a descriptor whose name pointer is NULL.

Prototype:

```
int
ORM_AttributeListCreate(
        ORM_NodeDef                   parent,        /* in */
        ORM_AppHandleDef              handle,        /* in */
        ORM_AspectDescrDef            aspectdescr,   /* in */
        ORM_AttributeDescrListDef     attrdescrlist, /* in */
        long                          attrcount      /* in */
        );
```

Parameters:

parent Pointer to an existing component node, who is the parent node of all newly created attribute nodes.

handle Pointer to the application instance, all attribute belongs to or ORM_HandleInherit (-1), which indicates, that the actual handle is determined by the parent (which again may have its handle set to ORM_HandleInherit!)

aspectdescr No description

attrdescrlist Pointer to an array of ORM_AttributeDescr, with name=NULL in the last element if *attrcount* is < 0.

attrcount	The number of attribute descriptors in the list or the number of initial attributes from this list to attach to this node or *-1*, if the end of the list (array) should be determined by a NULL nodeinfo pointer.

3.5 ORM Attribute Conversion Support

This section was generated from *<stdin>* by CDOC on Sun Jan 29 18:13:38 1995.

This part or the ORM Server Support Layer provides functions for converting generic ORM data types between their native (binary) and the ORM (ASCII) presentation. The interface between the attribute and the conversion layer is defined by to function types, one for converting application native data into an ORM representation, one to convert ORM attribute value strings into the applications native presentation. Beside the conversion functions provided by the ORM-SSL, every application may provide its own special converters as long as their interfaces conform these function types.

3.5.1 Function Type ORM_ConverterNativeToString

This function is called to convert a single native value into its string representation. In addition to the value string it may generate the range and unit strings, if the pointer values passed are non-null. If the converter function returns NULL in these pointers, the lower (attribute) layer may provide default strings if any.

Memory Allocation: The memory to hold the converted string value(s) has to be provided by the converter function. It is reasonable to use static memory for this purpose, because before the converter function is called again, the ORM protocol layer will copy the strings returned.

Declaration:

```
typedef ORM_Status (* ORM_ConverterNativeToStringFunc)(
                ORM_AppDataPtrDef          ptr,            /* in */
                size_t                     size,           /* in */
                ORM_AttributeDescrListDef datatype,        /* in */
                ORM_AppConverterArgDef     convarg,        /* in */
                ORM_String                 *strvalue,      /* out */
                ORM_String                 *strrange,      /* out */
                ORM_String                 *strunit        /* out */
                );
```

Fields:

ptr Address of native data element (e.g. attribute value)

size Byte-size of data element

datatype One of the ORM_AttributeTypes identifying the type of the native data element and its mapping to an ORM Protocol data type (??is this overloaded ??)

convarg Any kind of argument (pointer) for this converter (as provided with the attribute descriptor for ex.)

strvalue Where to store the pointer to the converted value string.

strrange Where to store the reference to the optional range string.

strunit Where to store the reference to the optional unit string.

3.5.2 Function Type ORM_ConverterStringToNative

This function is called to convert a single ORM string value into its native presentation. The pointer for the result usually points into a set of different attributes, e.g. an aspect, which usually is a (partial) copy of some application data instance.

Memory Allocation: The destination pointer provided references some valid memory (e.g. an aspect), but for references (the native value is a C-string for ex.), there is usually not enough space for the referenced value. This space must be allocated/provided by the converter itself. It is legal, to reference the original string as passed in to the converter function, but then the AspectCallSet function should make a copy, if the string is needed beyond this call.

Declaration:

```
typedef ORM_Status (* ORM_ConverterStringToNativeFunc)(
            ORM_AppDataPtrDef        dest,          /* in */
            size_t                   size,          /* in */
            ORM_AttribTypeDef        datatype,      /* in */
            ORM_AppConverterArgDef   convarg,       /* in */
            ORM_String               strvalue       /* in */
            );
```

Fields:

dest Address/destination of native data element (e.g. attribute value)

maxsize Maximum byte-size of data element

datatype One of the ORM_AttributeTypes identifying the type of the native data element

convarg Any kind of data (pointer) for this converter as provided with the attribute descriptor

strvalue The new attribute value in its ascii presentation.

returns ORM_ENoError if conversion was successfull and the resulting attribute value is valid or ORM_ERangeError.

3.5.3 ORM Built In Conversion Functions

The following functions are provided to convert generic C datatypes between their ORM and their native presentation. In addition sub functions are provided to support the special ORM SELECT and MCHOICE types, which are called by the generic converters. Along with these sets two new data structure (types) are introduced.

3.5.4 Function ORM_GenericNativeToString

This function converts standard C-data types into their ASCII presentation. It returns only the converted value, but does not support the range and unit parts (e.g. returns NULL for those, if requested). In case of SELECT or MCHOICE functions, this function calls the related ORM_Select.. or ORM_MChoice functions.

Note: It is currently open, whether the conversion argument *convarg* may be used to specify a format string a la *printf*. Furthermore it is currently open, whether a NULL conversion function in the attribute descriptor should be directed to this (default) function.

Arguments as for ORM_ConverterNativeToString!

Prototype:

```
ORM_Status ORM_GenericNativeToString(
          ORM_AppDataPtrDef         ptr,             /* in */
          size_t                    maxsize,         /* in */
          ORM_AttribTypeDef         type,            /* in */
          ORM_AppConverterArgDef    convarg,         /* in */
          ORM_String                *strvalue,       /* out */
          ORM_String                *rangevalue,     /* out */
          ORM_String                *strunit         /* out */
          );
```

No parameter descriptions are available.

3.5.5 Function ORM_GenericStringToNative

This function converts ASCII C-strings into standard C-datatypes. In case of SELECT or MCHOICE functions, this function calls the related ORM_Select.. or ORM_MChoice functions.

Note: It is currently open, whether the conversion argument *convarg* may be used to specify a format string a la *sscanf*. Furthermore it is currently open, whether a NULL conversion function in the attribute descriptor should be directed to this (default) function.

Arguments as for ORM_ConverterStringToNative!

Prototype:

```
ORM_Status ORM_GenericStringToNative(
          ORM_AppDataPtrDef         ptr,             /* in */
          size_t                    maxsize,         /* in */
          ORM_AttribTypeDef         type,            /* in */
          ORM_AppConverterArgDef    convarg,         /* in */
          ORM_String                strvalue         /* in */
          ).
```

No parameter descriptions are available.

3.5.6 Structure ORM_StringMapDef

This type of structure is used to map strings to binary values and vice versa. It may be used to convert internal flags and states to *friendly* names. StringMaps must be terminated by an entry with *name* set to NULL.

Declaration:

```
typedef struct ORM_StringMapTag {
        ORM_String   name;
        ORM_Key      key;
        } *ORM_StringMapDef;
```

Fields:

name Friendly name for this key.

key The binary native value of the key

3.5.7 ORM_StringMapToString

This function maps a value key to a string using the given StringMap. It returns the string of that map entry, whose key is equal to the given key, else it returns the string passed in *notfound*.

Prototype:

```
ORM_String
ORM_StringMapToString(  ORM_StringMapDef     map,
                ORM_Key           key,
                ORM_String        notfound);
```

Parameters:

map Pointer to a sequence of map entries

key Binary key value.

notfound string to give back, if none of the keys in the map matched.

3.5.8 ORM_StringMapToKey

This function maps a string value to a binary key using the given StringMap. It returns the key of that map entry, whose string is equal to the given key, else it returns the key passed in *invalidkey*.

Prototype:

```
ORM_Key
ORM_StringMapToKey( ORM_StringMapDef     map,
                ORM_String        name,
                ORM_Key           invalidkey);
```

Parameters:

map Pointer to a sequence of map entries

name No description

invalidkey No description

3.5.9 Structure ORM_StateMapDef

This structure is used to map *states* into strings, where a *state* is assumed to have a distinct set of possible next states, depending on the current value. E.g. this structure can be used to derive the set of possible new values i.e. it can provide the *range* value for a state attribute.

Otherwise it is used similar to the simpler StringMap structure. StateMaps must be terminated by an entry with *name* set to NULL.

Declaration:

```
typedef struct ORM_StateMapTag { /* not of the U.S.A.!! */
     ORM_String;  name;
     ORM_Key      state;
     ORM_String   validnexts;
     } *ORM_StateMapDef;
```

Fields:

name Friendly name for this key.

state The binary native value of this state

validnexts String of comma separated names of next valid states which may follow this state.

3.5.10 ORM_StateMapToString

Convert an encoding of a state into a *friendly* name using the given statemap. If the state could not be found, the string passed in *notfound* is returned.

Prototype:

```
ORM_Str...;
ORM_StateMapToString (  ORM_StateMapDef map,
                ORM_Key       state,
                ORM_String    notfound);
```

Parameters:

map Pointer to a (name=NULL) terminated state map.

state the binary state

notfound string to return, if none of the entries in the map had exactly the given state key.

3.5.11 ORM_StateMapToKey

Convert a string representation of a state into a native encoding of a *state using the given statemap*. If the string could not be found, the state passed in *invalidstate* is returned.

Prototype:

```
ORM_Key
ORM_StateMapToKey ( ORM_StateMapDef map,
                    ORM_String  name,
                    ORM_Key     invalidstate);
```

Parameters:

map	Pointer to a (name=NULL) terminated state map.
name	No description
invalidstate	No description

3.5.12 ORM_StateMapNextByKey

Return the comma separated list of valid next states given the current state.

Prototype:

```
ORM_String
ORM_StateMapNextByKey(  ORM_StateMapDef map,
                ORM_String  state);
```

Parameters:

map	Pointer to a (name=NULL) terminated state map.
state	the binary state

3.6 ORM Dump & Restore Support

This section was generated from *<stdin>* by CDOC on Fri Jan 27 19:59:34 1995.

This module of the ORM Server Support Library supports the dump and restore of complete subtrees, and therefore can be used to save the current configuration to a persistant storage media (i.e. the MSF Warehouse) and reload it from there. The actual IO functions are currently not supported by this layer or the support library at all!

Dump and Restore are functions of the ORM SSL and not of the ORM protocol (i.e. there is no *DUMP* or *RESTORE* request defined in the protocol).

This implies, that these functions have to be dispatched out of the application layer explicitly. One (intended) way to dispatch those functions interactively is to provide pseudo components in every subtree, which should be independent storable/reloadable. These contain the required parameters like Warehouse location or version name as attributes. An Attribute-Set request to this subtree then results in the execution of the corresponding function.

Under the layered view of the ORM SSL, these two functions belong to the protocol layer, as they use (nearly) the same functionality of the higher layers via upcalls.

3.6.1 General Model:

Starting from a given node, which is used as the root of the relevant subtree to dump, all components, object links and writable attributes with their meta information are recursivly extracted relative to the current subtree root. The extended/meta information on the persistent media can be used to interprete the stored attributes and apply changes to the stored version without the ORM server/application alive but through special clients (by an ORM/Warehouse gateway for example).

The dumped ORM tree can be used to reload the whole subtree at any time, by providing the node and call the *restore* function of the ORM SSL (which is a special kind of Set-Request).

This special kind of SET request creates a new situation, as components (or any new subtree) may have been created dynamically by the ORM server application on request. On the next cold start of the application, these subtrees do not exist.

This results in failed lookup requests by the ORM protocol layer, which usually is treated as an error (remember: ORM-P has no direct support for object/component creation, but this is emulated by sets of writeonly attributes in separate subtrees, i.e. *New..*). To handle this case, the application can provide a special function during application context setup to create new instances including the ORM subtree (ORM_NodeNotFoundTrapFunc()).

A parameter is passed to this creation function, which indicates, whether this situation was caused by a regular ORM protocol request or by an internally generated *restore* request, so the application code can still decide to refuse the creation.

3.6.2 ORM_Dump

This function extracts the ORM entities in the subtree pointed to by *subtree* into the character buffer, so it can be used by a later ORM_Restore function (or can be used as a subrequest in a regular ORM protocol request).

It is the responsibility of the caller to provide a sufficient buffer, which can hold the subtree information of the given depth!

Prototype:

```
ORM_Status
ORM_Dump(     ORM_AppNodeDef   subtree,
              long             depth,
              long             what,
              ORM_String       buffer,
              long             *maxlen
              );
```

Parameters:

subtree The root of the subtree to dump. All navigation information is saved relative to this node.

depth The depth, up to which entities in this subtree should be extracted. A depth of 0 means, direct childs of the given sub-root only, i.e. if the subtree points to a component node with an attribute node as one of its direct children, the name of the attribute would be extracted, but not the value or other extended attribute information, if depth=0. A depth of -1 extracts the whole subtree, independent of its depth.

what Is a bitmask, defining what kind of entities should be extracted:
ORM_DumpSetObjects ORM_DumpSetComponents
ORM_DumpSetAttributes ORM_DumpSetWritable
ORM_DumpSetDefault = Objects | Components | Writable
ORM_DumpSetEveryThing = Objects | Components | Attributes

buffer The address of a character buffer, where to store the extracted entity information

maxlen Pointer to the maximum length of this buffer. On return, maxlen will contain the number of bytes used in this buffer including the C-String '\0' terminator.

3.6.3 ORM_Restore

Function to reload the saved ORM information into an existing subtree, where at least the root of the given subtree has to exist. *Note:* Because the restore request may fail with some attribute modifications already performed, an application may want to call ORM_Dump (into a temporary buffer) before actually calling ORM_Restore, to be able to *undo* the partial operations.

Prototype:

```
ORM_Status
ORM_Restore(        ORM_AppHandler   subtree,
                    ORM_String       buffer,
                    long             *length
                    );
```

Parameters:

subtree Node of the subtree to load the management information into.

buffer pointer to ORM subrequest sequence.

length pointer to length of the request. On return, this will contain the number of bytes processed from this request.

3.7 ORM SSL Generic Datatypes

```
#ifndef _ORM_TYPE_H
#define _ORM_TYPE_H

/*
 * Some generic definitions, may become obsolete
 */

typedef enum {
    False,
    True
    } boolean;

#ifndef NULL
#define NULL (void *)0
#endif

typedef unsigned long size_t;

#define ORM_Ptr( base, offset)  (void *)((size_t )(base)+(size_t)(offset))

#define ORM_Malloc(x)     (void *)malloc(x)
#define ORM_Free(x)       free(x)

/*
 * more ORM specific stuff
 */

/*
 * How requests and responses are passed to the ORM protocol layer
 */

typedef char     *ORM_RequestDef;

typedef char     *ORM_ResponseDef;

/*
 * The principal type of every ORM protocol entity, e.g. names and values,
 * but also used most C-strings.
 */

typedef char     *ORM_String;

/*
 * Used for StateMaps and String Maps as the lookup key
 */

typedef long     ORM_Key;


/*
 * The following are various opaque handles. Opaque mainly to the protocol
 * layer but also for the lower of two stacked layers.
 */

typedef void     *ORM_AppNodeDef;
typedef void     *ORM_AppHandleDef;
typedef void     *ORM_AppAspectDef;
typedef void     *ORM_AppDataPtrDef;
typedef void     *ORM_AppAttribDescrDef;
typedef void     *ORM_AppCallContextDef;
typedef void     *ORM_AppConverterArgDef;

/*
 * Valid Access modes for an attribute
 */

typedef enum {
     ORM_AttribModeNone,
     ORM_AttribModeRW,
     ORM_AttribModeRO,
     ORM_AttribModeWO,
     ORM_AttribModeRWP
     } ORM_AttribModeDef;

/*
 * Known (native) datatypes, which are supported by the Generic converter
 */

typedef enum {
     ORM_AttribTypeNone,
     ORM_AttribTypeInt1,
     ORM_AttribTypeUInt1,
     ORM_AttribTypeInt2,
     ORM_AttribTypeUInt2,
     ORM_AttribTypeInt4,
     ORM_AttribTypeUInt4,
     ORM_AttribTypeInt8,
     ORM_AttribTypeUInt8,
     ORM_AttribTypeReal32,
     ORM_AttribTypeReal64,
     ORM_AttribTypeString,
     ORM_AttribTypeHexOct,
     ORM_AttribTypeSelect,      /* 1 out of many */
     ORM_AttribTypeState,       /* 1 out of many, but with dynamic range */
     ORM_AttribTypeOption,      /* binary switch ON/OFF YES/NO */
     ORM_AttribTypeMChoice,     /* n out of many */
     ORM_AttribTypeUnknown
     } ORM_AttribTypeDef;

/*
 * ORM Error Codes, used as well by the protocol as by the ORM SSL
 */

typedef enum {
   ORM_ENoError,              /* Operation successfull! */
   ORM_EPermission,           /* None or wrong auth.information */
   ORM_ENoSuchNode,           /* some name in pathname could not be found */
   ORM_ENoSuchAttribute,      /* Attribute in Set-Request doesn't exist */
   ORM_ENoSuchObject,         /* Object/Manager could not be found */
   ORM_EInvalidOperation,     /* Operation not applicable to node type */
   ORM_EProtocol,             /* ORM protocol violation */
   ORM_ECommunication,        /* lower level comm error */
   ORM_ERange,                /* new attribute value out of range */
   ORM_EParameterList,        /* set of attributes not applicable */
   ORM_EMissingAttribute,     /* mandatory attrib. missing or NULL */
   ORM_ENoSpace,              /* internal allocation */
   ORM_ENoBuffer,             /* response buffer */
```

```
    ORM_EInternal,              /* ORM Internal error -> bug */
    ORM_EApplication,           /* application level error -> bug */
    } ORM_Status;


/*
 * Types of nodes in the virtual tree. Note, that only nodes of type
 * ORM_NodeTypeComponent can have children!
 */

typedef enum {
     ORM_NodeTypeUnknown,
     ORM_NodeTypeObject,
     ORM_NodeTypeComponent,
     ORM_NodeTypeAttribute,
     ORM_NodeTypeAny
     } ORM_NodeTypeDef;

/*
 * Types of ORM requests. Note that the Dump and Restore requests are
 * not part of the ORM protocol, but only available within the ORM
 * server support library
 */

typedef enum {
     ORM_RequestObjectGet,
     ORM_RequestComponentGet,
     ORM_RequestAttributeGet,
     ORM_RequestAttributeInfoGet,
     ORM_RequestAttributeSet,
     ORM_RequestSet,
     ORM_RequestGet,
     ORM_RequestDump,
     ORM_RequestRestore
     } ORM_RequestTypeDef;


#endif
```

WHAT IS CLAIMED IS:

1. A system for managing objects, including a first server, comprising:

a first receiver portion configured to receive a request in a hypermedia format;

a first translator portion configured to convert the hypermedia request to an object request;

a sender portion configured to send the object request to an object manager;

a second receiver portion configured to receive a response from the object manager; and

a second translator portion configured to convert the object manager response to the hypermedia format.

2. The system of claim 1, further comprising a second server, including:

a third receiver portion configured to receive a request in a hypermedia format;

a third translator portion configured to convert the hypermedia request to an object request;

a second sender portion configured to send the object request to an object manager;

a fourth receiver portion configured to receive a response from the object manager; and

a fourth translator portion configured to convert the object manager response to the hypermedia format.

3. The system of claim 1, further comprising:

a second sending portion configured to send the hypermedia format data from the sender portion to a browser to be displayed.

4. The system of claim 1, where the object manager manages a self-describing object.

5. The system of claim 1, where the object manager manages a non-self-describing object.

6. The system of claim 5, where the object manager performs a "worm" function.

7. A method for browsing objects, where a browser communicates with a server, comprising the steps, performed by the browser, of:

sending an initial URL to the server;

receiving first data from the server, where the first data specifies an object corresponding to the URL;

sending user-entered data associated with the object to the server; and

receiving second data from the server, where the second data specifies a second object corresponding to the user-entered data.

8. The method of claim 7,

wherein the step of sending an initial URL to the server comprises the step of sending an initial URL known to the browser, where the URL is the URL of the server.

9. The method of claim 7,

wherein the step of sending an initial URL to the server comprises the step of sending an initial URL entered by the user, where the URL is the URL of the server.

10. The method of claim 7,

wherein the step of sending user-entered data associated with the object to the server includes the step of indicating a "set" operation in the user-entered data.

11. The method of claim 7,

wherein the step of sending user-entered data associated with the object to the server includes the step of indicating a "get" operation in the user-entered data.

12. The method of claim 7, wherein the step of receiving second data from the server includes the step of receiving data corresponding to an attribute value of the object.

13. The method of claim 7, wherein the step of receiving second data from the server includes the step of receiving data corresponding to a second object linked to the first object via an object-link.

14. A computer program product comprising:

a computer usable medium having computer readable code embodied therein for managing objects, the computer program product comprising:

computer readable program code devices configured to cause a computer to effect receiving a request in a hypermedia format;

computer readable program code devices configured to cause a computer to effect converting the hypermedia request to an object request;

computer readable program code devices configured to cause a computer to effect sending the object request to an object manager;

computer readable program code devices configured to cause a computer to effect receiving a response from the object manager; and

computer readable program code devices configured to cause a computer to effect converting the object manager response to a second hypermedia format.

15. The computer program product of claim 14, further comprising:

computer readable program code devices configured to cause a computer to effect sending the second hypermedia format data to a browser to be displayed.

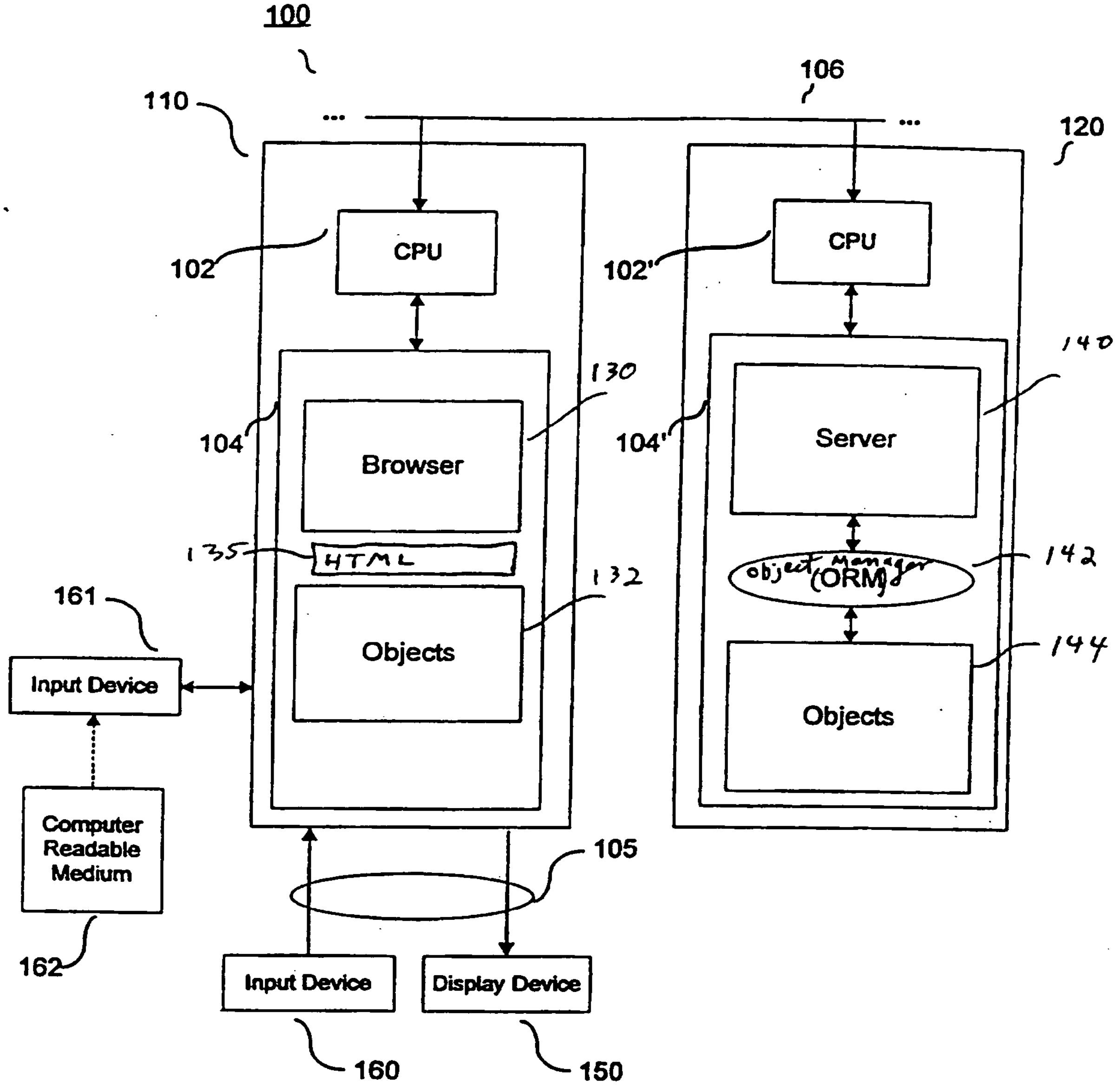

100
106
110
120
CPU
CPU
102
102'
104
104'
Browser
130
Server
140
135
HTML
object Manager (ORM)
142
Objects
132
Objects
144
161
Input Device
Computer Readable Medium
162
105
Input Device
Display Device
160
150

Fig. 1

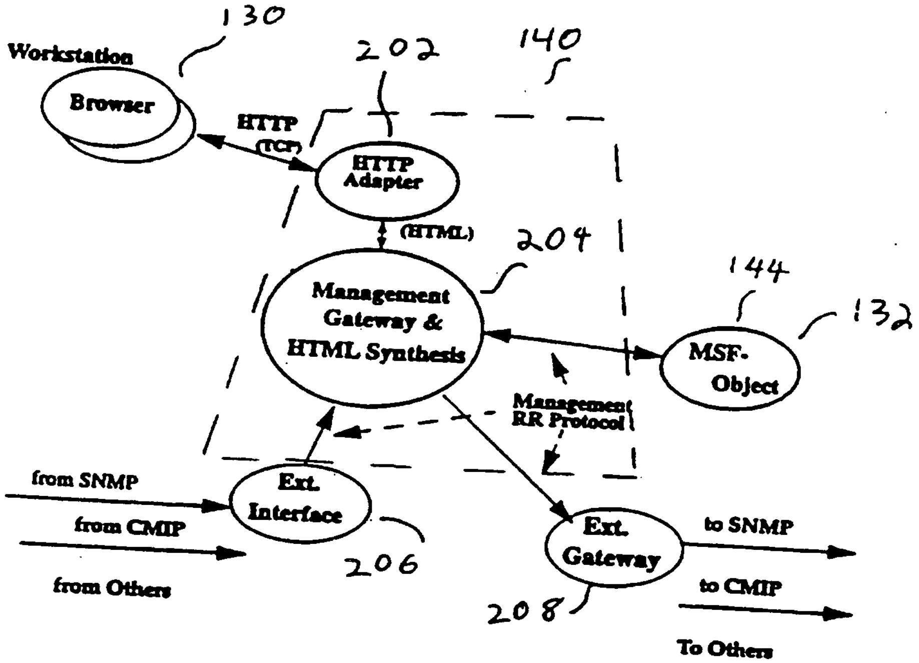

130
Workstation
Browser
202
140
HTTP
(TCP)
HTTP
Adapter
(HTML)
204
Management
Gateway &
HTML Synthesis
144
132
MSF-
Object
Management
RR Protocol
from SNMP
from CMIP
from Others
Ext.
Interface
206
Ext.
Gateway
208
to SNMP
to CMIP
To Others

Fig. 2

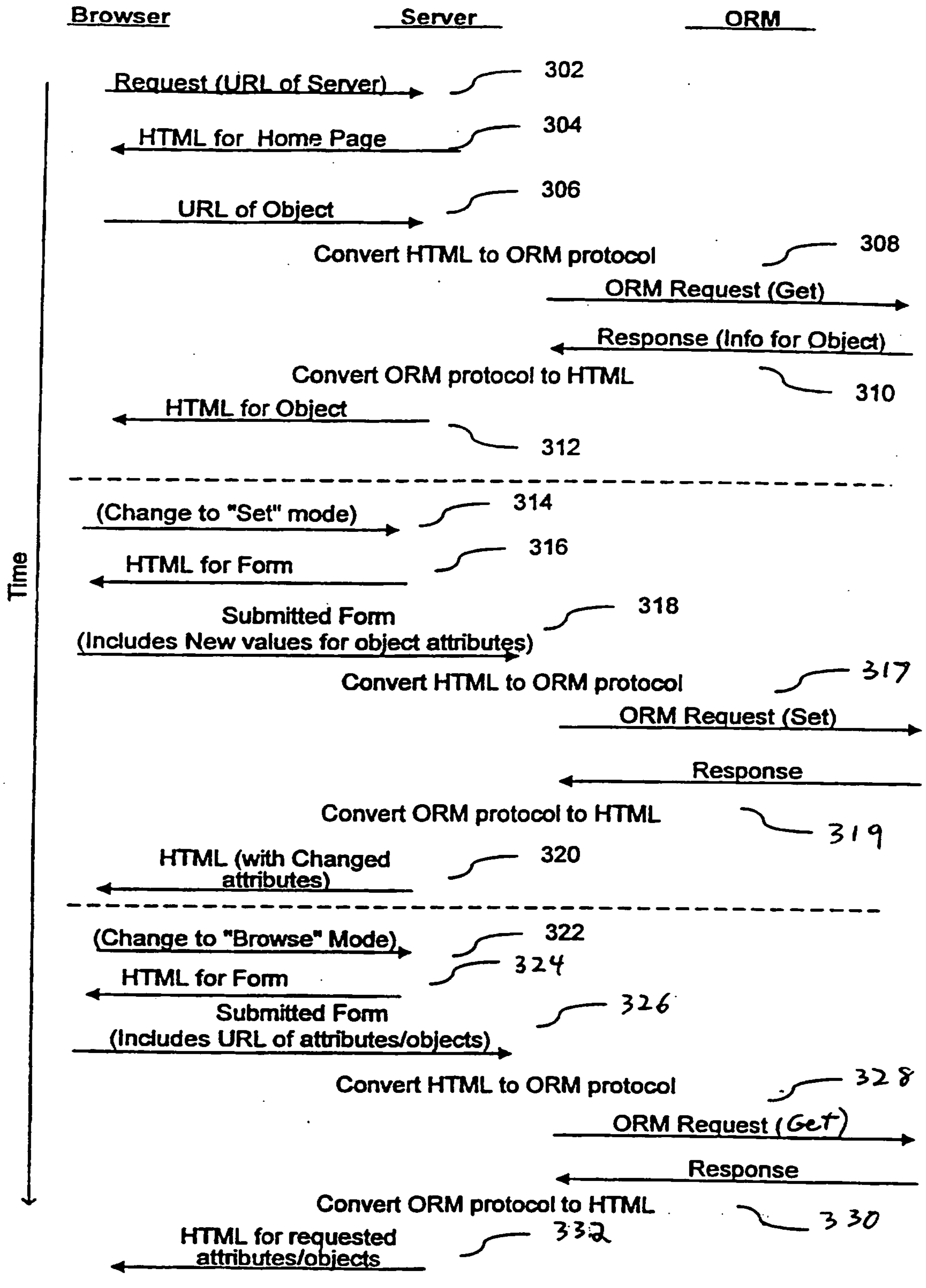

Browser
Server
ORM
Request (URL of Server)
302
HTML for Home Page
304
URL of Object
306
Convert HTML to ORM protocol
308
ORM Request (Get)
Response (Info for Object)
Convert ORM protocol to HTML
310
HTML for Object
312
(Change to "Set" mode)
314
HTML for Form
316
Submitted Form
(Includes New values for object attributes)
318
Convert HTML to ORM protocol
317
ORM Request (Set)
Response
Convert ORM protocol to HTML
319
HTML (with Changed attributes)
320
(Change to "Browse" Mode)
322
HTML for Form
324
Submitted Form
(Includes URL of attributes/objects)
326
Convert HTML to ORM protocol
328
ORM Request (Get)
Response
Convert ORM protocol to HTML
330
HTML for requested attributes/objects
332
Time

Fig. 3

A System as a Tree of Managed Entities:

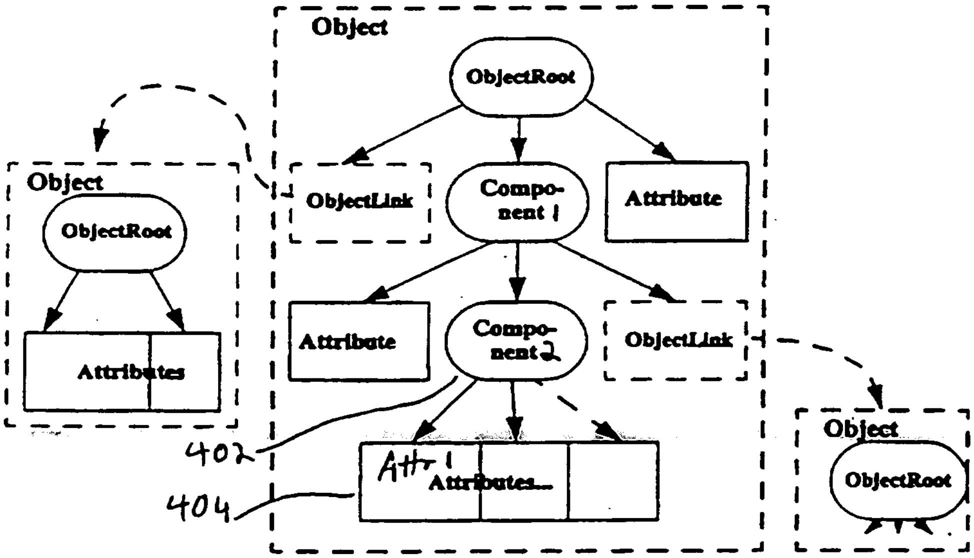
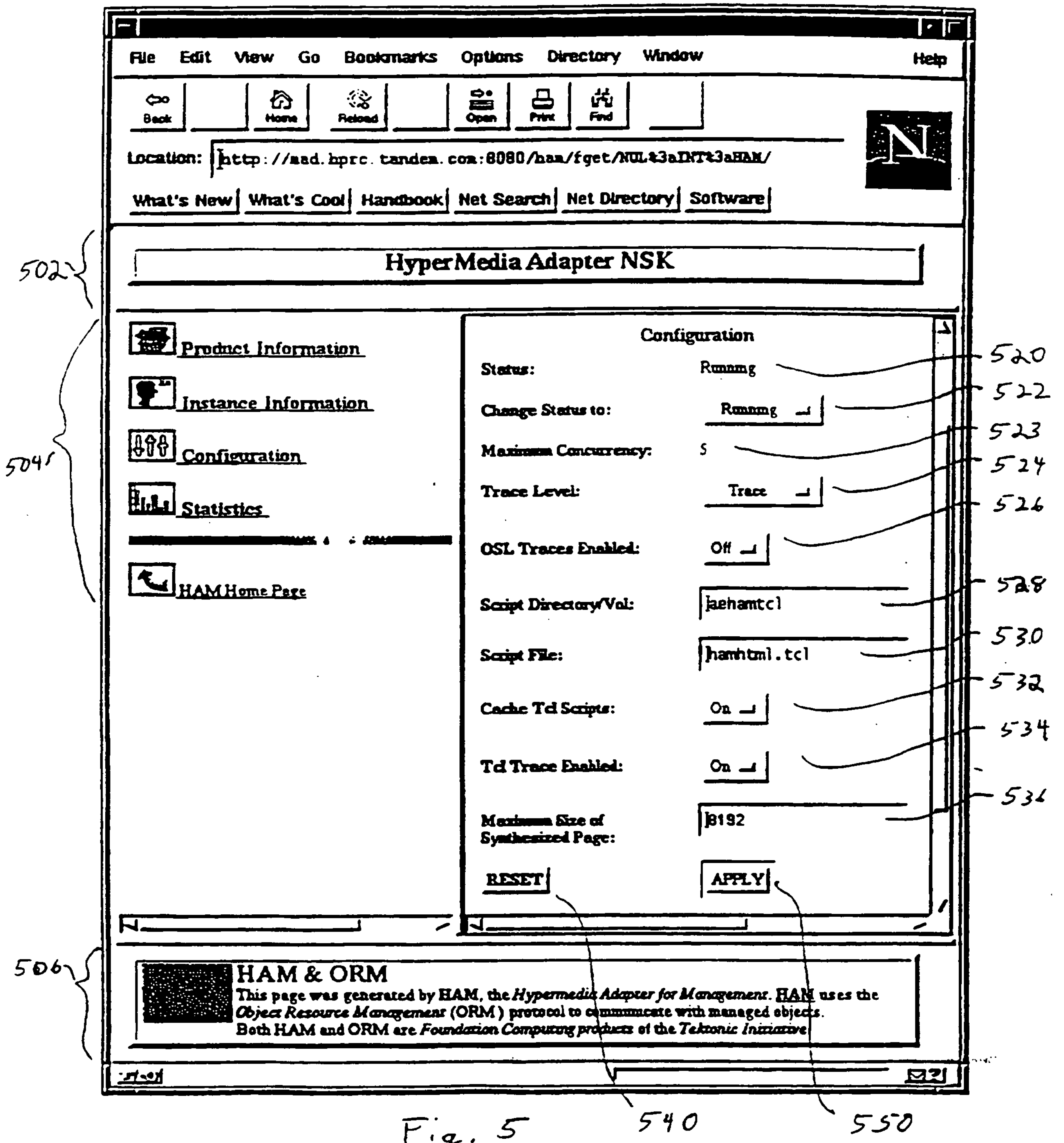


Fig. 4



Location: http://aad.hprc.tandem.com:8080/ham/fget/NUL%3aINT%3aHAM/

HyperMedia Adapter NSK

Configuration

Status: Running

Change Status to: Running

Maximum Concurrency: 5

Trace Level: Trace

OSL Traces Enabled: Off

Script Directory/Vol: aehamtcl

Script File: hamhtml.tcl

Cache Tcl Scripts: On

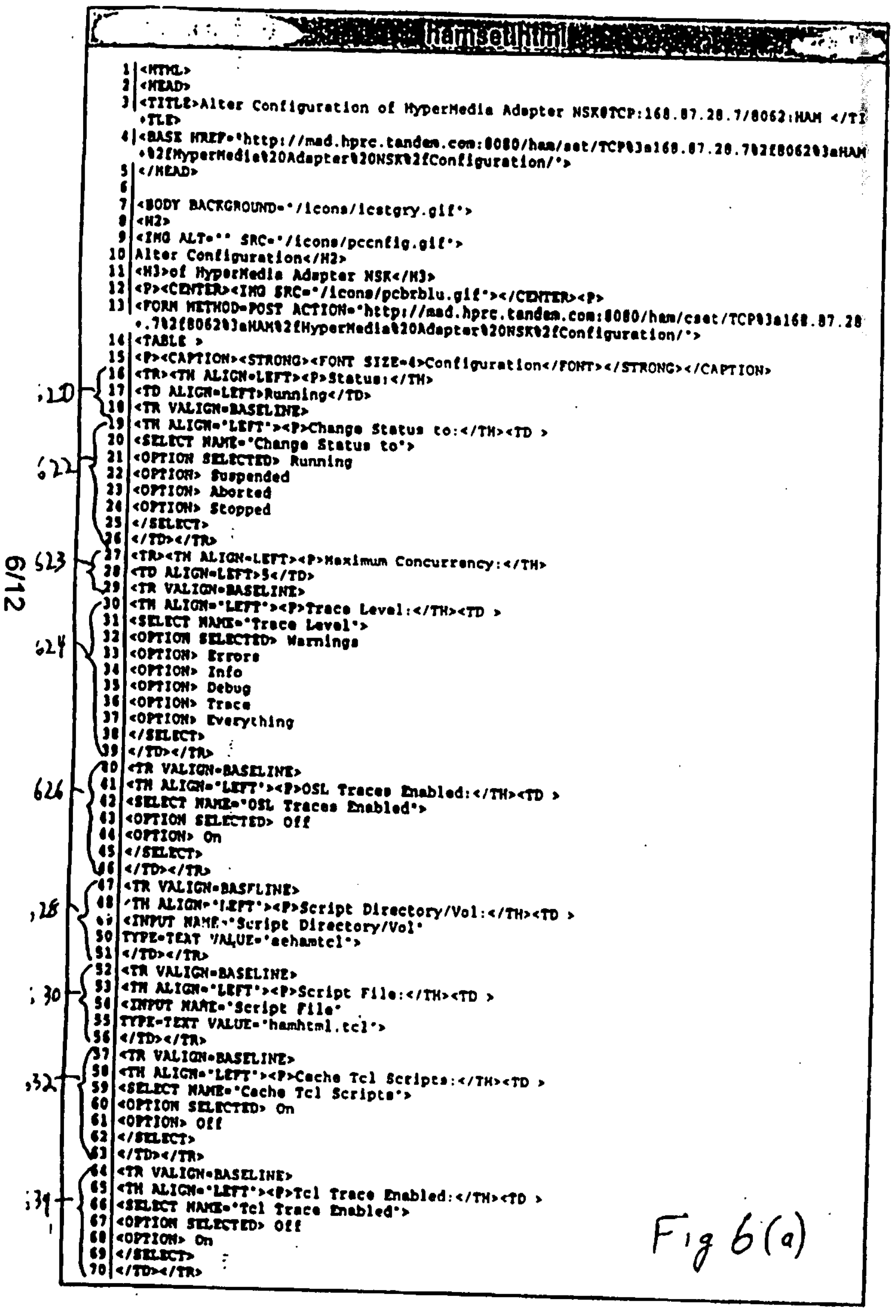
Tcl Trace Enabled: On

Maximum Size of Synthesized Page: 8192

HAM & ORM

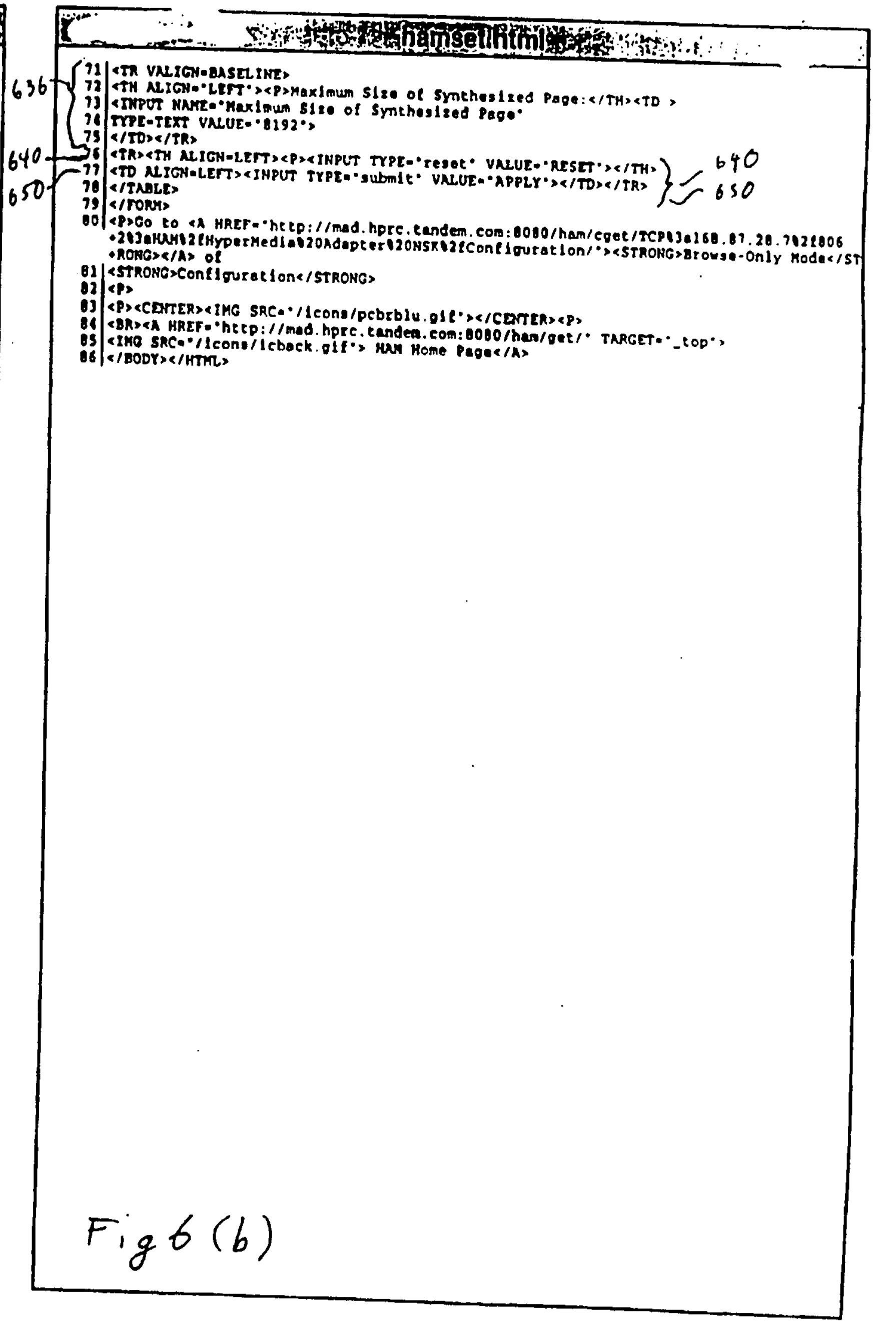
This page was generated by HAM, the *Hypermedia Adapter for Management*. HAM uses the *Object Resource Management* (ORM) protocol to communicate with managed objects. Both HAM and ORM are *Foundation Computing products* of the *Tektonic Initiative*

Fig. 5



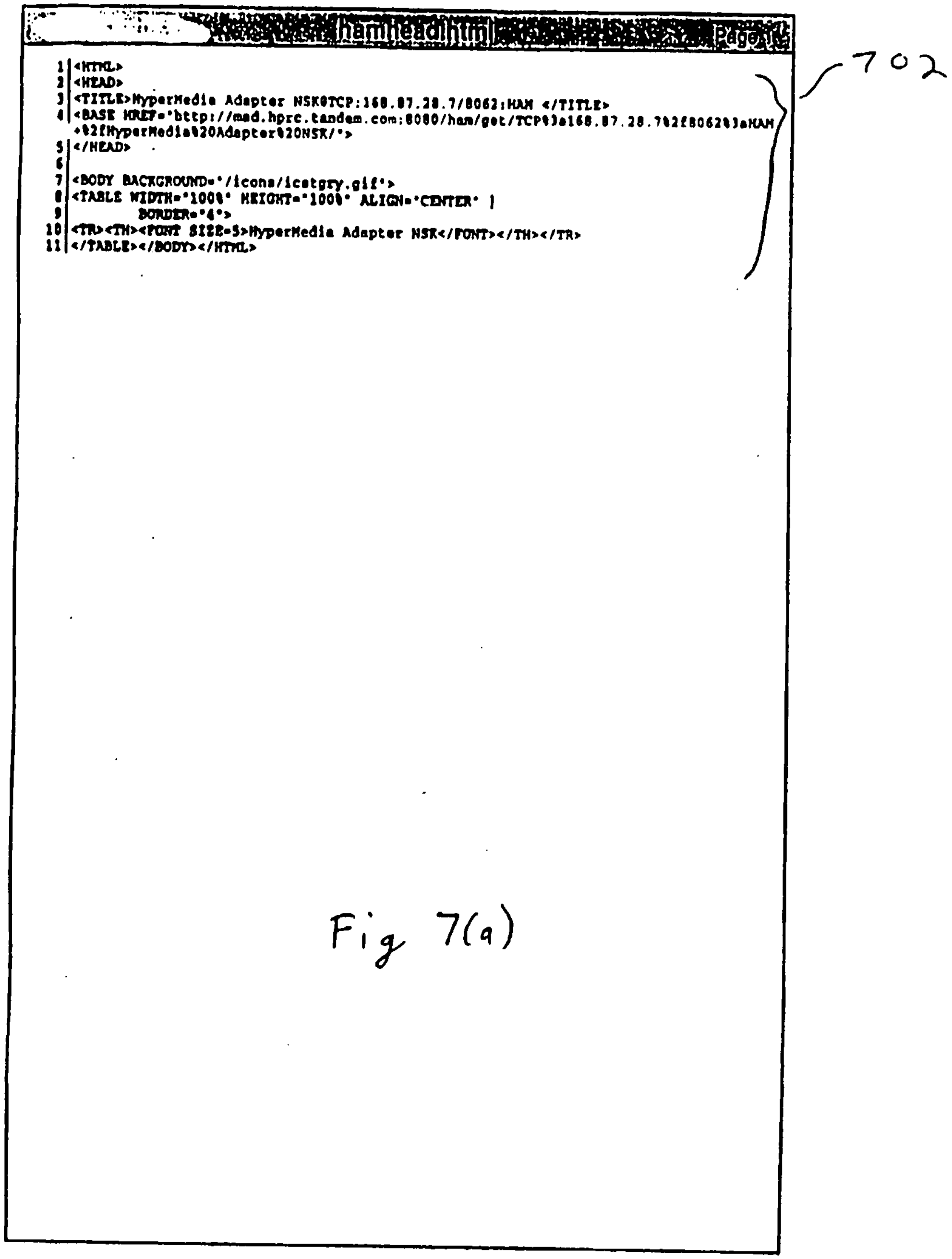
```
<HTML>
<HEAD>
<TITLE>Alter Configuration of HyperMedia Adapter NSK@TCP:168.87.28.7/8062:HAM </TI
•TLE>
<BASE HREF='http://mad.hprc.tandem.com:8080/ham/set/TCP%3a168.87.28.7%2f8062%3aHAM
•%2fHyperMedia%20Adapter%20NSK%2fConfiguration/'>
</HEAD>

<BODY BACKGROUND='/icons/icstgry.gif'>
<H2>
<IMG ALT='' SRC='/icons/pccnfig.gif'>
Alter Configuration</H2>
<H3>of HyperMedia Adapter NSK</H3>
<P><CENTER><IMG SRC='/icons/pcbrblu.gif'></CENTER><P>
<FORM METHOD=POST ACTION='http://mad.hprc.tandem.com:8080/ham/cset/TCP%3a168.87.28
•.7%2f8062%3aHAM%2fHyperMedia%20Adapter%20NSK%2fConfiguration/'>
<TABLE >
<P><CAPTION><STRONG><FONT SIZE=4>Configuration</FONT></STRONG></CAPTION>
<TR><TH ALIGN=LEFT><P>Status:</TH>
<TD ALIGN=LEFT>Running</TD>
<TR VALIGN=BASELINE>
<TH ALIGN='LEFT'><P>Change Status to:</TH><TD >
<SELECT NAME='Change Status to'>
<OPTION SELECTED> Running
<OPTION> Suspended
<OPTION> Aborted
<OPTION> Stopped
</SELECT>
</TD></TR>
<TR><TH ALIGN=LEFT><P>Maximum Concurrency:</TH>
<TD ALIGN=LEFT>5</TD>
<TR VALIGN=BASELINE>
<TH ALIGN='LEFT'><P>Trace Level:</TH><TD >
<SELECT NAME='Trace Level'>
<OPTION SELECTED> Warnings
<OPTION> Errors
<OPTION> Info
<OPTION> Debug
<OPTION> Trace
<OPTION> Everything
</SELECT>
</TD></TR>
<TR VALIGN=BASELINE>
<TH ALIGN='LEFT'><P>OSL Traces Enabled:</TH><TD >
<SELECT NAME='OSL Traces Enabled'>
<OPTION SELECTED> Off
<OPTION> On
</SELECT>
</TD></TR>
<TR VALIGN=BASFLINE>
/TH ALIGN='LEFT'><P>Script Directory/Vol:</TH><TD >
<INPUT NAME='Script Directory/Vol'
TYPE=TEXT VALUE='eehamtcl'>
</TD></TR>
<TR VALIGN=BASELINE>
<TH ALIGN='LEFT'><P>Script File:</TH><TD >
<INPUT NAME='Script File'
TYPE=TEXT VALUE='hamhtml.tcl'>
</TD></TR>
<TR VALIGN=BASELINE>
<TH ALIGN='LEFT'><P>Cache Tcl Scripts:</TH><TD >
<SELECT NAME='Cache Tcl Scripts'>
<OPTION SELECTED> On
<OPTION> Off
</SELECT>
</TD></TR>
<TR VALIGN=BASELINE>
<TH ALIGN='LEFT'><P>Tcl Trace Enabled:</TH><TD >
<SELECT NAME='Tcl Trace Enabled'>
<OPTION SELECTED> Off
<OPTION> On
</SELECT>
</TD></TR>
```

Fig 6(a)



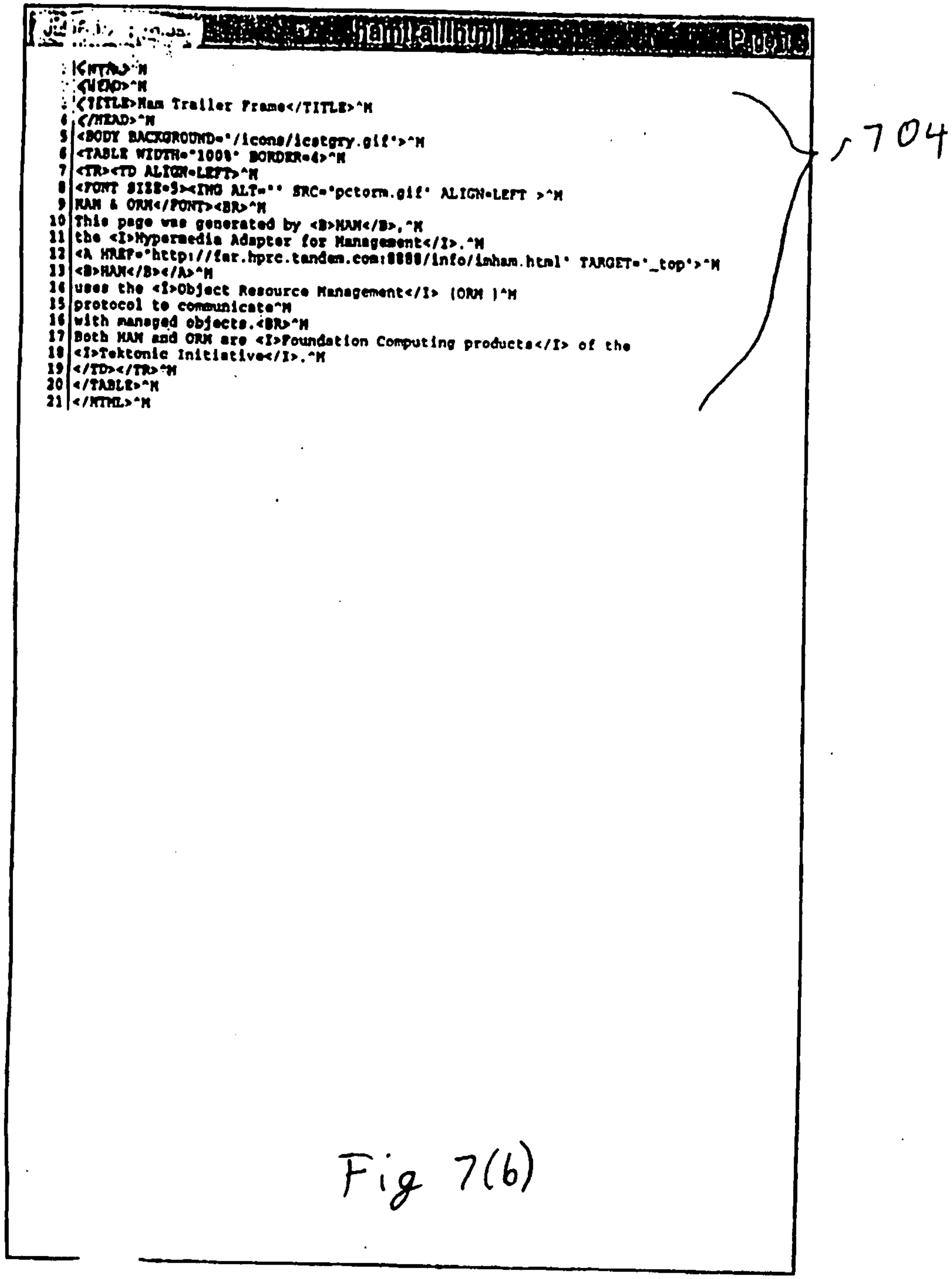
```
<TR VALIGN=BASELINE>
<TH ALIGN='LEFT'><P>Maximum Size of Synthesized Page:</TH><TD >
<INPUT NAME='Maximum Size of Synthesized Page'
TYPE=TEXT VALUE='8192'>
</TD></TR>
<TR><TH ALIGN=LEFT><P><INPUT TYPE='reset' VALUE='RESET'></TH>
<TD ALIGN=LEFT><INPUT TYPE='submit' VALUE='APPLY'></TD></TR>
</TABLE>
</FORM>
<P>Go to <A HREF='http://mad.hprc.tandem.com:8080/ham/cget/TCP%3a168.87.28.7%2f806
•2%3aHAM%2fHyperMedia%20Adapter%20NSK%2fConfiguration/'><STRONG>Browse-Only Mode</ST
•RONG></A> of
<STRONG>Configuration</STRONG>
<P>
<P><CENTER><IMG SRC='/icons/pcbrblu.gif'></CENTER><P>
<BR><A HREF='http://mad.hprc.tandem.com:8080/ham/get/' TARGET='_top'>
<IMG SRC='/icons/icback.gif'> HAM Home Page</A>
</BODY></HTML>
```

Fig 6 (b)



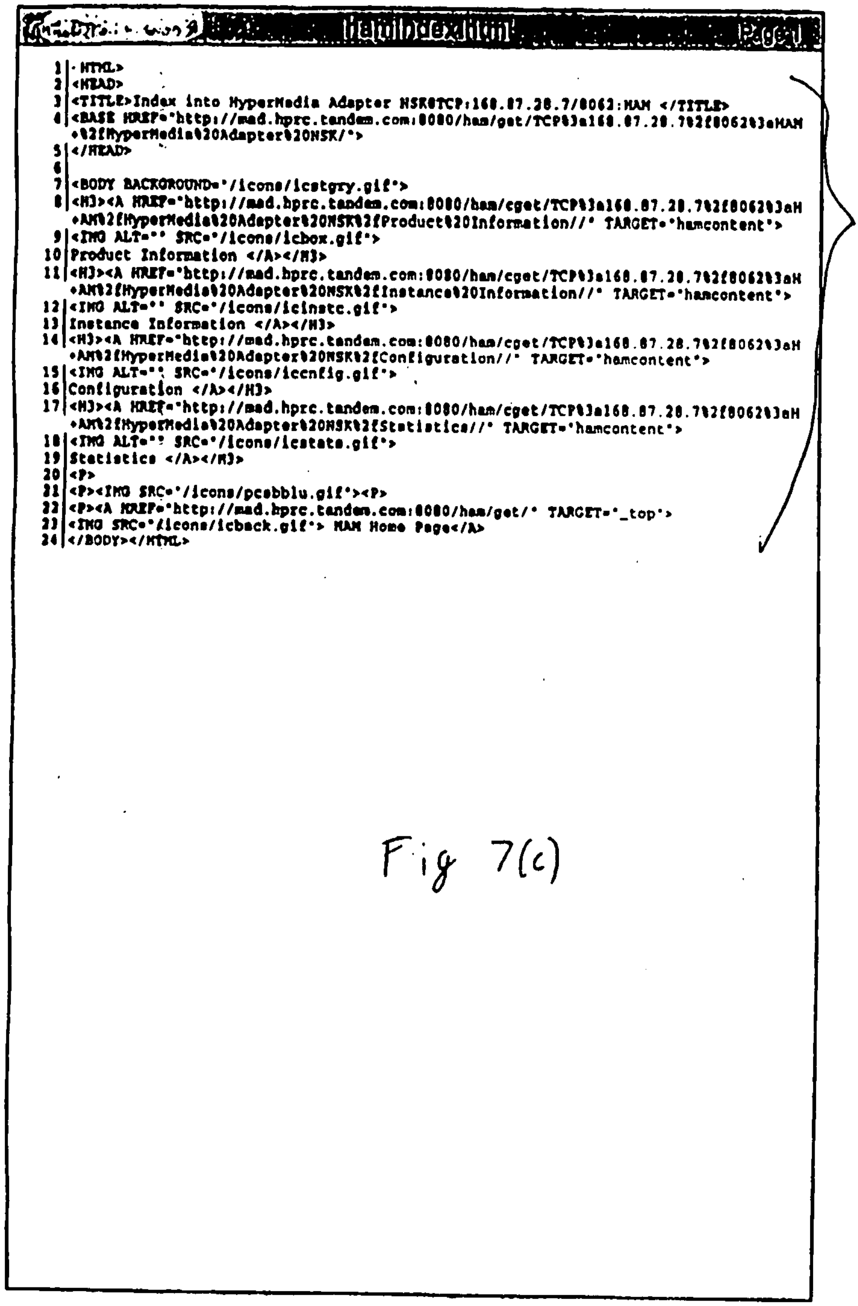
```
<HTML>
<HEAD>
<TITLE>HyperMedia Adapter NSK@TCP:168.87.28.7/8062:HAM </TITLE>
<BASE HREF='http://mad.hprc.tandem.com:8080/ham/get/TCP%3a168.87.28.7%2f8062%3aHAM
+%2fHyperMedia%20Adapter%20NSK/'>
</HEAD>

<BODY BACKGROUND='/icons/icstgry.gif'>
<TABLE WIDTH='100%' HEIGHT='100%' ALIGN='CENTER' |
        BORDER='4'>
<TR><TH><FONT SIZE=5>HyperMedia Adapter NSK</FONT></TH></TR>
</TABLE></BODY></HTML>
```

Fig 7(a)



```
<HTML>^M
<HEAD>^M
<TITLE>Ham Trailer Frame</TITLE>^M
</HEAD>^M
<BODY BACKGROUND='/icons/icstgry.gif'>^M
<TABLE WIDTH='100%' BORDER=4>^M
<TR><TD ALIGN=LEFT>^M
<FONT SIZE=5><IMG ALT='' SRC='pctorm.gif' ALIGN=LEFT >^M
HAM & ORM</FONT><BR>^M
This page was generated by <B>HAM</B>,^M
the <I>Hypermedia Adapter for Management</I>.^M
<A HREF='http://far.hprc.tandem.com:8888/info/imham.html' TARGET='_top'>^M
<B>HAM</B></A>^M
uses the <I>Object Resource Management</I> (ORM )^M
protocol to communicate^M
with managed objects.<BR>^M
Both HAM and ORM are <I>Foundation Computing products</I> of the
<I>Tektonic Initiative</I>.^M
</TD></TR>^M
</TABLE>^M
</HTML>^M
```

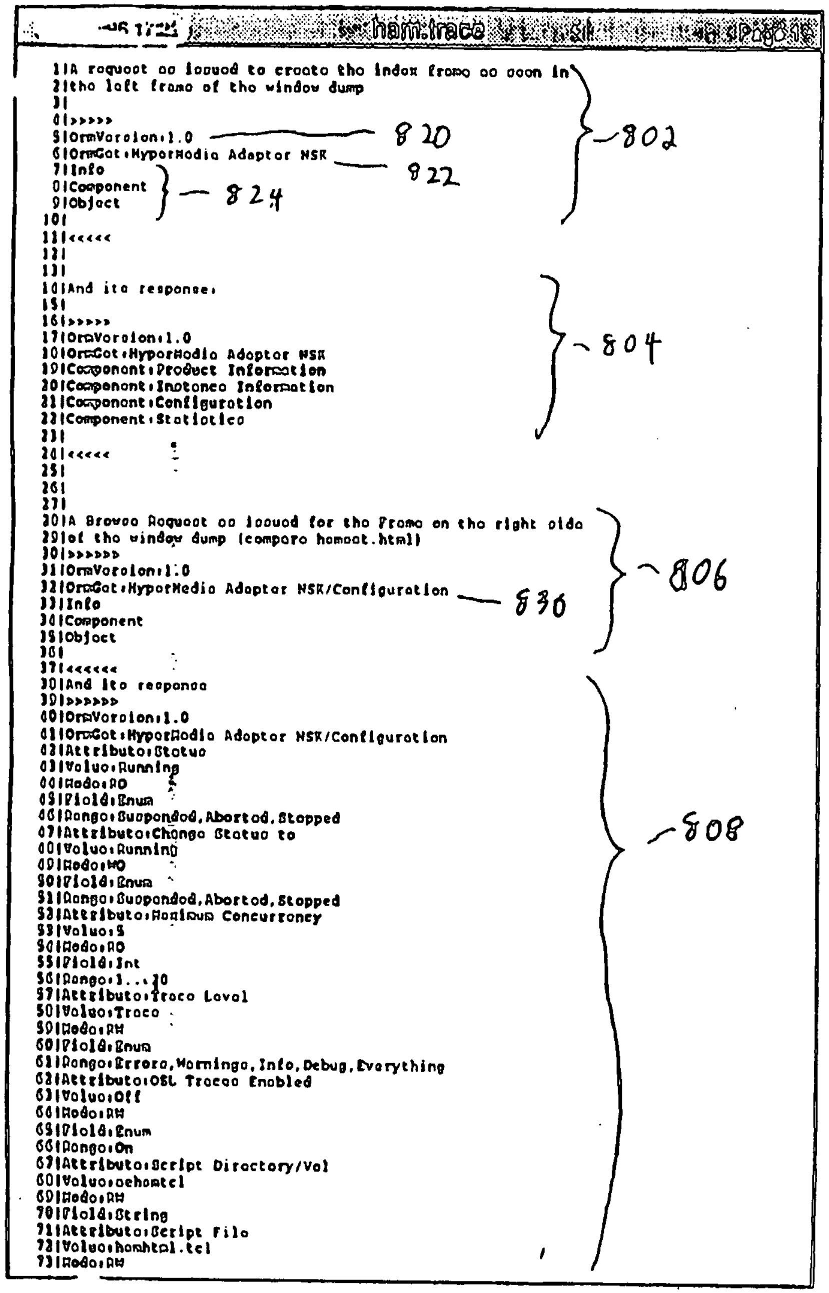
Fig 7(b)



```
<HTML>
<HEAD>
<TITLE>Index into HyperMedia Adapter NSK@TCP:168.87.28.7/8062:HAM </TITLE>
<BASE HREF="http://mad.hprc.tandem.com:8080/ham/get/TCP%3a168.87.28.7%2f8062%3aHAM
+%2fHyperMedia%20Adapter%20NSK/">
</HEAD>

<BODY BACKGROUND="/icons/icstgry.gif">
<H3><A HREF="http://mad.hprc.tandem.com:8080/ham/cget/TCP%3a168.87.28.7%2f8062%3aH
+AM%2fHyperMedia%20Adapter%20NSK%2fProduct%20Information//" TARGET="hamcontent">
<IMG ALT="" SRC="/icons/icbox.gif">
Product Information </A></H3>
<H3><A HREF="http://mad.hprc.tandem.com:8080/ham/cget/TCP%3a168.87.28.7%2f8062%3aH
+AM%2fHyperMedia%20Adapter%20NSK%2fInstance%20Information//" TARGET="hamcontent">
<IMG ALT="" SRC="/icons/icinstc.gif">
Instance Information </A></H3>
<H3><A HREF="http://mad.hprc.tandem.com:8080/ham/cget/TCP%3a168.87.28.7%2f8062%3aH
+AM%2fHyperMedia%20Adapter%20NSK%2fConfiguration//" TARGET="hamcontent">
<IMG ALT="" SRC="/icons/iccnfig.gif">
Configuration </A></H3>
<H3><A HREF="http://mad.hprc.tandem.com:8080/ham/cget/TCP%3a168.87.28.7%2f8062%3aH
+AM%2fHyperMedia%20Adapter%20NSK%2fStatistics//" TARGET="hamcontent">
<IMG ALT="" SRC="/icons/icstats.gif">
Statistics </A></H3>
<P>
<P><IMG SRC="/icons/pcebblu.gif"><P>
<P><A HREF="http://mad.hprc.tandem.com:8080/ham/get/" TARGET="_top">
<IMG SRC="/icons/icback.gif"> HAM Home Page</A>
</BODY></HTML>
```

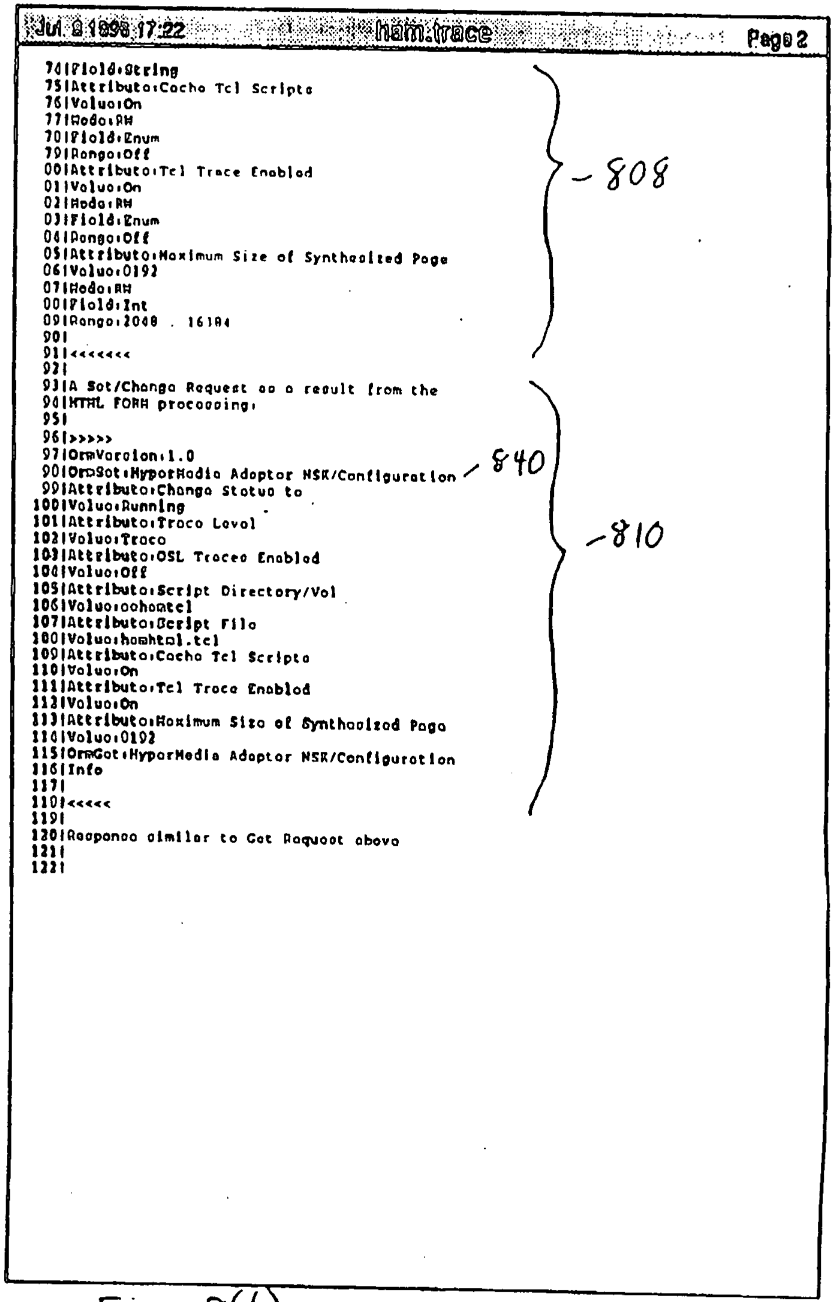
Fig 7(c)

ORM Protocol examples



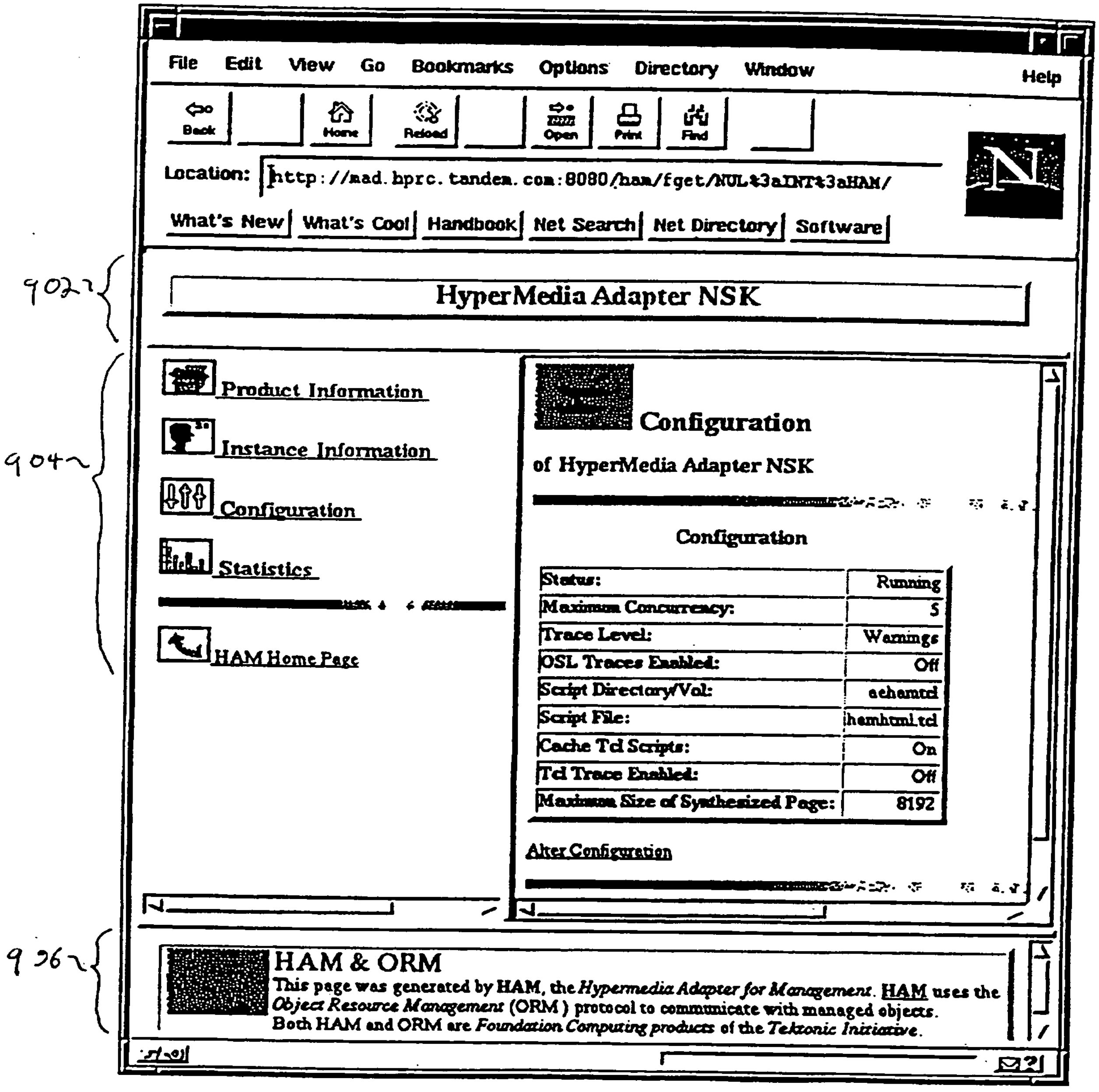
```
A request as issued to create the index frame as soon in
the left frame of the window dump

>>>>>
OrmVersion:1.0
OrmGet:HyperMedia Adaptor NSR
Info
Component
Object

<<<<<


And its response:

>>>>>
OrmVersion:1.0
OrmGet:HyperMedia Adaptor NSR
Component:Product Information
Component:Instance Information
Component:Configuration
Component:Statistics

<<<<<



A Browse Request as issued for the Frame on the right side
of the window dump (compare homeat.html)
>>>>>>
OrmVersion:1.0
OrmGet:HyperMedia Adaptor NSR/Configuration
Info
Component
Object

<<<<<<
And its response
>>>>>>
OrmVersion:1.0
OrmGet:HyperMedia Adaptor NSR/Configuration
Attribute:Status
Value:Running
Mode:RO
Field:Enum
Range:Suspended,Aborted,Stopped
Attribute:Change Status to
Value:Running
Mode:WO
Field:Enum
Range:Suspended,Aborted,Stopped
Attribute:Maximum Concurrency
Value:5
Mode:RO
Field:Int
Range:1...10
Attribute:Trace Level
Value:Trace
Mode:RW
Field:Enum
Range:Errors,Warnings,Info,Debug,Everything
Attribute:OSL Traces Enabled
Value:Off
Mode:RW
Field:Enum
Range:On
Attribute:Script Directory/Vol
Value:sehomtcl
Mode:RW
Field:String
Attribute:Script File
Value:homhtml.tcl
Mode:RW
```

Fig. 8(a)



```
Field:String
Attribute:Cache Tcl Scripts
Value:On
Mode:RW
Field:Enum
Range:Off
Attribute:Tcl Trace Enabled
Value:On
Mode:RW
Field:Enum
Range:Off
Attribute:Maximum Size of Synthesized Page
Value:8192
Mode:RW
Field:Int
Range:2048 . 16384

<<<<<<<

A Set/Change Request as a result from the
HTML FORM processing:

>>>>>
OrmVersion:1.0
OrmSet:HyperMedia Adaptor NSR/Configuration
Attribute:Change Status to
Value:Running
Attribute:Trace Level
Value:Trace
Attribute:OSL Traces Enabled
Value:Off
Attribute:Script Directory/Vol
Value:sehomtcl
Attribute:Script File
Value:homhtml.tcl
Attribute:Cache Tcl Scripts
Value:On
Attribute:Tcl Trace Enabled
Value:On
Attribute:Maximum Size of Synthesized Page
Value:8192
OrmGet:HyperMedia Adaptor NSR/Configuration
Info

<<<<<

Response similar to Get Request above


```

Fig. 8(b)



http://mad.hprc.tandem.com:8080/ham/fget/NUL%3aINT%3aHAM/

HyperMedia Adapter NSK

Product Information

Instance Information

Configuration

Statistics

HAM Home Page

Configuration

of HyperMedia Adapter NSK

Configuration

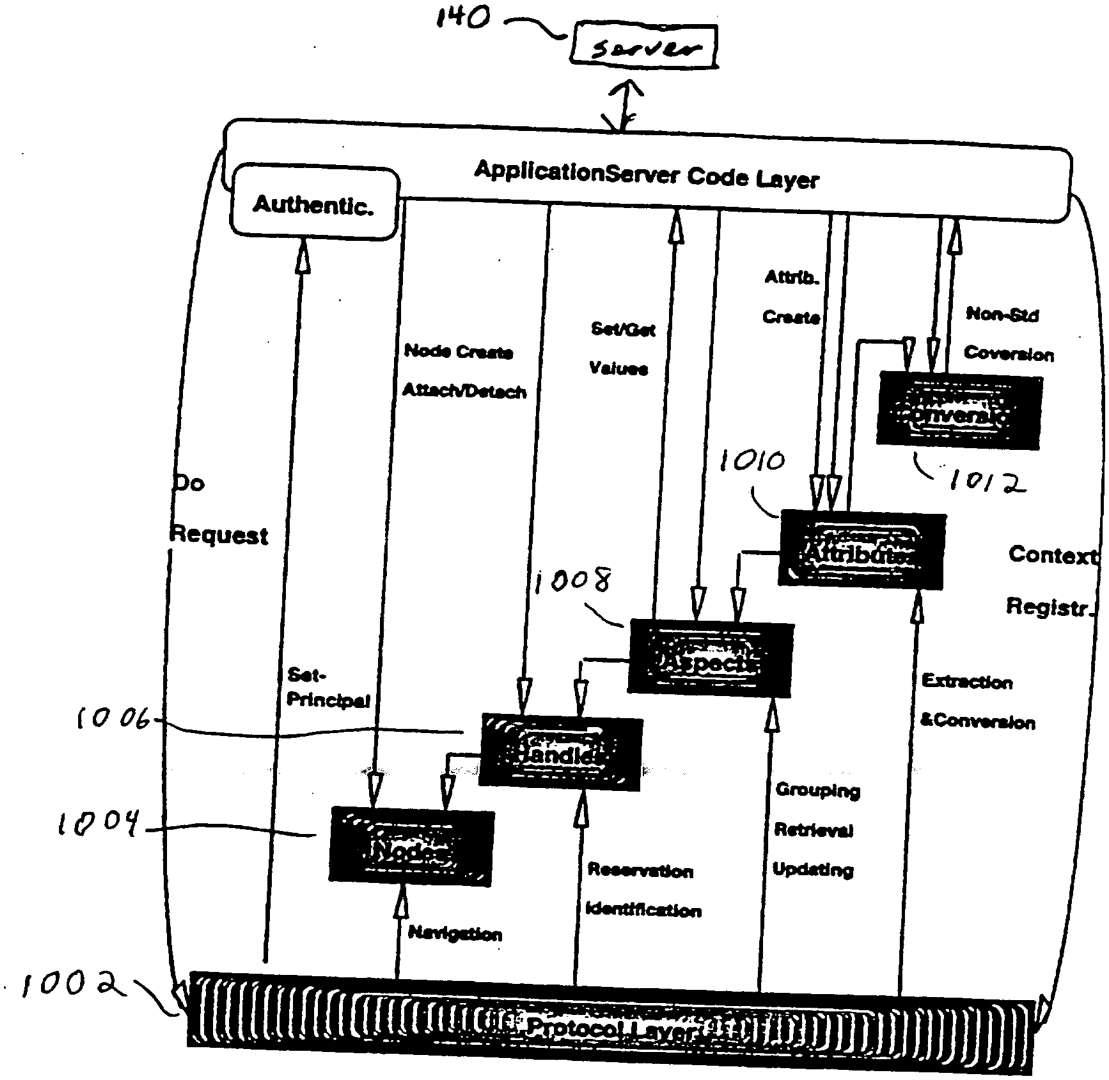
Status:	Running
Maximum Concurrency:	5
Trace Level:	Warnings
OSL Traces Enabled:	Off
Script Directory/Vol:	achamtcl
Script File:	hamhtml.tcl
Cache Tcl Scripts:	On
Tcl Trace Enabled:	Off
Maximum Size of Synthesized Page:	8192

Alter Configuration

HAM & ORM

This page was generated by HAM, the *Hypermedia Adapter for Management*. HAM uses the *Object Resource Management* (ORM) protocol to communicate with managed objects. Both HAM and ORM are *Foundation Computing products* of the *Tektonic Initiative*.

Fig. 9



The Layered Structure of the ORM Support Library

Fig. 10

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 97/11885

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JAGANNATHAN V ET AL: "COLLABORATIVE INFRASTRUCTURES USING THE WWW AND CORBA-BASED ENVIRONMENTS" PROCEEDINGS - THE WORKSHOP ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, 19 June 1996, pages 292-297, XP000645510 see page 293, column 1, line 39 - page 294, column 1, line 5 -----	1,14

□ Further documents are listed in the continuation of box C.

□ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search 6 November 1997	Date of mailing of the international search report 12. 11. 97
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Katerbau, R

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

WEST Search History

DATE: Friday, January 24, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=OR			
L16	L14 not L15	55	L16
L15	L14 and (gateway or gate$way)	10	L15
L14	L13 not L10	65	L14
L13	L12 and ((707/$ or 709/$)!.ccls.)	71	L13
L12	L6 and L11	334	L12
L11	plural$3 near3 (database$ or template$ or container$ or folder$3)	28599	L11
L10	L8 and ((707/$ or 709/$).ccls.)	6	L10
L9	L8 and gate$way	4	L9
L8	L6 and L7	78	L8
L7	plural$3 near3 (container$ or folder$3)	22150	L7
L6	plural$3 near3 (register$3)	21112	L6
L5	L3 and ((container or folder$) with history)	7	L5
L4	L3 and gateway	20	L4
L3	L1 and L2	552	L3
L2	plural$3 with (register$3)	38977	L2
L1	plural$3 with (container$ or folder$3)	42649	L1

END OF SEARCH HISTORY

PATENT

IN THE UNITED STATES

PATENT AND TRADEMARK OFFICE

APPLICANT:	Michael De Angelo
APPLICATION NO.:	09/284,113
FILING DATE:	April 7, 1999
TITLE:	System And Method For Creating And Manipulating Information Containers With Dynamic Registers
EXAMINER:	Not yet assigned
GROUP ART UNIT:	2771
ATTY. DKT. NO.:	21114-03726

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner For Patents, Washington, D.C. 20231, on the date shown below:

Dated: 1/16/03 By: Greg T. Sueoka

COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

REQUEST TO WITHDRAW AS ATTORNEYS

SIR:

Pursuant to 37 CFR §§ 1.36 and 10.40, the attorneys of record listed below, on whose behalf the undersigned attorney has signed this request and on whose behalf the undersigned attorney is authorized to sign, respectfully request permission to withdraw as attorneys of record in the above-referenced patent application, because the client has failed to pay one or more bills rendered by the undersigned's law firm for an unreasonable period of time.

☒ There is currently no outstanding official action known to the undersigned attorney dated prior to the date of this Request to Withdraw for which a response is now due to be filed.

□ There is an outstanding term for response to an official action that is set to expire on ________. The above term is extendible under 37 CFR § 1.136(a), until ______________.

□ The client has been notified of the official action and of the deadline to respond to the official action.

□ A response to the outstanding official action is attached.

□ A petition for extension of time is requested and the fees for the extension are/are not being submitted.

The withdrawing attorneys of record are:

☒ associated with Customer Number 00758 →

OR

□ as named below:

Name	Registration Number	Name	Registration Number
Greg T. Sueoka	33,800	James Okamoto	40,110

The attorneys of record listed above, on whose behalf the undersigned attorney has signed this request and on whose behalf the undersigned attorney is authorized to sign, respectfully request that all future correspondence regarding this application be addressed to the last know address:

Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

A copy of this Request is being sent to Michael De Angelo, Information Equity Corporation, 100 South Sunrise Boulevard, Suite 470, Palm Springs, CA 92262 at the last-known address. This Request to Withdraw is being submitted in triplicate.

Respectfully submitted,
Michael De Angelo

Dated: 1/16/03

By: Greg Su
Greg T. Sueoka, Reg. No.: 33,800
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
Tel.: (650) 858-7194
Fax.: (650) 494-1417

WEST Search History

DATE: Tuesday, January 28, 2003

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
DB=USPT; PLUR=YES; OP=OR			
L18	L15 and ("OOP" or "object-oriented")	9	L18
L17	L16 not L13	37	L17
L16	L15 and (register$3 with (data or file or resource$ or information) with (container$ or folder$3))	39	L16
L15	L10 and L14	555	L15
L14	plural$ with (container$ or folder$3)	42715	L14
L13	L12 not L8	43	L13
L12	L11 and ("OOP" or "object-oriented")	45	L12
L11	L9 and L10	1065	L11
L10	plural$ with register$3	39067	L10
L9	plural$ with (container$ or folder$3 or database$)	50485	L9
L8	L3 and L7	17	L8
L7	L6 not L4	43	L7
L6	L1 and ("OOP" or "object-oriented")	44	L6
L5	L2 and ("OOP" or "object-oriented")	1	L5
L4	L2 and L3	35	L4
L3	(container$ or folder$3) with (Id or ids or identif$5)	6737	L3
L2	plural$ with (container$ or folder$3) with register$3	292	L2
L1	(container$ or folder$3) with register$3	2703	L1

END OF SEARCH HISTORY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 02/11/2003

GREG T SUEOKA
FENWICK & WEST
TWO PALO ALTO SQUARE
PALO ALTO, CA 94306

EXAMINER	
NGUYEN, CAM LINH T	
ART UNIT	PAPER NUMBER
2171	

DATE MAILED: 02/11/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 07-01)

Office Action Summary	**Application No.** 09/284,113	**Applicant(s)** DE ANGELO, MICHAEL	
	Examiner Cam-Linh T. Nguyen	**Art Unit** 2171	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 April 1999* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *07 April 1999* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *3 & 5* .

4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

DETAILED ACTION

Specification

1. This application does not contain an abstract of the disclosure as required by 37 CFR 1.72(b). An abstract on a separate sheet is required.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1 - 36 are rejected under 35 U.S.C. 102(e) as being anticipated by McKeehan et al (U.S. 6,016,495).

♦ As per claim 20,

McKeehan et al. discloses a method for creating an interactive information container comprising:

- "Forming a container" See Fig. 8, element 870, Fig. 10, col. 18 line 33 – 47.
- "Selecting an interactive register for the container" See Fig. 2B, col. 11 line 17 – 27, col. 20 line 59 – 67, Fig. 17A col. 24 line 65 – col. 25 line 9.
- "Identifying an item for inclusion in a container" corresponds to a particular resource or object that need to register and stored in the storage.

- "Creating a container element that includes the identified item" See Fig. 2B col. 11 line 17 – 27.when an item is registered and stored in the persistent storage, it is considered as an identified item, and the storage that includes the identified item is called "container" as in Fig. 10.

♦ As per claim 1,29,

With all limitations as in claim 20 further claim 1 includes:

- "Plurality of containers" See Fig. 1, 2A, 8, col. 15 line 47 – col. 16 line 40, col. 17 line 24 – 37.
- "Plurality of registers" See Fig. 2A.
- "A first register storing a unique container identification value" See fig. 11, col. 19 line 3 – 8, 59 - 61.
- "The second register stores information and evolves according to the relationship, use and interaction of the container with other containers" See Fig. 11; col. 19 line 15 – 62.
- "A gateway attached to and forming part of the container... controlling the interaction of the container with other containers" corresponds to the Transaction Manager (See col. 20 line 63 – 67).

♦ As per claim 2,

- "The information element is one from the group of text, graphic...a system" corresponds to the text of the objects that registered in the container (See Fig. 2A).

♦ As per claim 3 - 4,

- "One container history register for storing information regarding past interaction of the container with other container... modified" See col. 20 line 6 – 13, col. 23 line 21 – 42.

♦ As per claim 5 - 7,

- "Plurality of registers include at least one predefined register" corresponds to the methods that are defined by the system (See col. 23 line 21 – 42).
- "Plurality of registers include a user created register" corresponds to the extensible methods that defined by user (See col. 23 line 21 – 42).

♦ As per claim 8 – 9, 31 – 32,

- "Plurality of registers ... controlling the relationship of the container with other containers" corresponds to the Lock register because the lock class controls about the concurrent of objects stored in the container (See col. 19 line 24 – 39).
- Because the system support locking manager, it must include an "active time, passive time, neutral time", which shows the interactive of an object with other objects.

♦ As per claim 10, 33,

- "Plurality of registers include at least one acquire register" See col. 21 line 17 – 21, col. 19 line 55 – 61.

♦ As per claim 11 – 14,

- Because the system 800 is virtual addressing mechanisms that allow the programs to access to other storage, therefore, the system must have register that referring the space or location of a container.

♦ As per claim 15 - 19, 35 – 36,

Because the Transaction Manager carries out the transaction control by interacting with objects of the extensible Resource class that are registered to it (Col. 20 line 67 – col. 21 line 13), therefore, it must including the means of "allowing interaction, gather information, reporting information, and including the rules defining the interaction of the container".

♦ As per claim 21 - 22, 30,

- "Displaying a plurality of container levels" See Fig. 12, col. 22 line 1 – 22.
- "Receiving input from user selecting one of the displayed container level" corresponds to the query of user to retrieve a particular object in a container.
- "Displaying a container template corresponding to the container level input" corresponds to the result of the query.

♦ As per claim 23,

Because each container store different data structure, the system must providing a data structure as part of the container element (See col. 1 line 36 – 40, col. 2 line 28 – 38).

♦ As per claim 24 - 26,

- In order for a container work properly, the container must have a gateway to control the interaction of the container with other containers. In the instance reference, the gateway is corresponding to the Transaction Manager (See col. 20 line 63 – 67), because the transaction manager comprises a list of transaction ID that controls all transactions between containers (col. 24 line 62 – 64). In addition, the container also has to determine the current gateway or register when a new container is created.

♦ As per claim 27 – 28, 34,

As noted above, the system creates some methods in order to create a new container; it must include the determination of available register to store items into the container (See col. 23 line 5 – 42).

Conclusion

1. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Itakura et al (U.S. 6,351,745) discloses a communication system for distributing such message as advertisement to user of terminal equipment.

2. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cam-Linh T. Nguyen whose telephone number is 703-305- 1951. The examiner can normally be reached on Monday - Friday from 8:00 am to 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Safet Metjahic, can be reached on (703) 308- 1436. The fax phone number for the organization where this application or proceeding is assigned is 703- 746- 7239.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703- 305-3900.

Cam-Linh Nguyen
Art Unit 2171

LN

SAFET METJAHIC
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Notice of References Cited	Application/Control No. 09/284,113	Applicant(s)/Patent Under Reexamination DE ANGELO, MICHAEL
	Examiner Cam-Linh T. Nguyen	Art Unit 2171
		Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,016,495	01-2000	McKeehan et al.	707/103R
	B	US-6,351,745	02-2002	Itakura et al.	707/10
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Sheet 1 of 1

FORM PTO-1449
(REV. 6-89)

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

INFORMATION DISCLOSURE CITATION

(Use several sheets if necessary)

Attorney's Docket No. 3726

Serial No. 09/284,113

Applicant Michael De Angelo

Filing Date April 7, 1999

Group Art Unit Unknown

U.S. PATENT DOCUMENTS

Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
LN	A	5 7 6 8 5 1 0	06/16/98	Gish	395	200.33	07/01/96
LN	B	5 8 4 8 2 4 6	12/08/98	Gish	395	200.58	07/01/96

FOREIGN PATENT DOCUMENTS

		Document Number	Date	Country	Class	Subclass	Translation Yes	Translation No

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER Nguyen Lam Lich

DATE CONSIDERED 1/30/03

EXAMINER: Initial if references considered, whether or not citation is in conformance with MPEP § 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Sheet 1 of 1

FORM PTO-1449 (REV. 6-89) U.S. DEPARTMENT OF COMMERCE Patent and Trademark Office **INFORMATION DISCLOSURE CITATION** (Use several sheets if necessary)	Attorney's Docket No. 3726	Serial No. 09/284,113
	Applicant Michael De Angelo	
	Filing Date April 7, 1999	Group Art Unit 2771

U.S. PATENT DOCUMENTS

Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
LN	C	5,664,208	09/02/97	Pavley et al.	395	777	May 16, 1995

FOREIGN PATENT DOCUMENTS

		Document Number	Date	Country	Class	Subclass	Translation	
							Yes	No
LN	D	WO 98 02831	01/22/98	PCT	G06F	17/30		

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER

DATE CONSIDERED 1/30/03

EXAMINER: Initial if references considered, whether or not citation is in conformance with MPEP § 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 10/30/2003

GREG T SUEOKA
FENWICK & WEST
TWO PALO ALTO SQUARE
PALO ALTO, CA 94306

EXAMINER	
NGUYEN, CAM LINH T	
ART UNIT	PAPER NUMBER
2171	7

DATE MAILED: 10/30/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

7

3

Notice of Abandonment	**Application No.** 09/284,113	**Applicant(s)** DE ANGELO, MICHAEL	
	Examiner Cam-Linh T. Nguyen	**Art Unit** 2171	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

This application is abandoned in view of:

1. ☒ Applicant's failure to timely file a proper reply to the Office letter mailed on *11 February 2003*.
   (a) □ A reply was received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the period for reply (including a total extension of time of _____ month(s)) which expired on _____.
   (b) □ A proposed reply was received on _____, but it does not constitute a proper reply under 37 CFR 1.113 (a) to the final rejection.
   (A proper reply under 37 CFR 1.113 to a final rejection consists only of: (1) a timely filed amendment which places the application in condition for allowance; (2) a timely filed Notice of Appeal (with appeal fee); or (3) a timely filed Request for Continued Examination (RCE) in compliance with 37 CFR 1.114).
   (c) □ A reply was received on _____ but it does not constitute a proper reply, or a bona fide attempt at a proper reply, to the non-final rejection. See 37 CFR 1.85(a) and 1.111. (See explanation in box 7 below).
   (d) ☒ No reply has been received.

2. □ Applicant's failure to timely pay the required issue fee and publication fee, if applicable, within the statutory period of three months from the mailing date of the Notice of Allowance (PTOL-85).
   (a) □ The issue fee and publication fee, if applicable, was received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the statutory period for payment of the issue fee (and publication fee) set in the Notice of Allowance (PTOL-85).
   (b) □ The submitted fee of $_____ is insufficient. A balance of $_____ is due.
   The issue fee required by 37 CFR 1.18 is $_____. The publication fee, if required by 37 CFR 1.18(d), is $_____.
   (c) □ The issue fee and publication fee, if applicable, has not been received.

3. □ Applicant's failure to timely file corrected drawings as required by, and within the three-month period set in, the Notice of Allowability (PTO-37).
   (a) □ Proposed corrected drawings were received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the period for reply.
   (b) □ No corrected drawings have been received.

4. □ The letter of express abandonment which is signed by the attorney or agent of record, the assignee of the entire interest, or all of the applicants.

5. □ The letter of express abandonment which is signed by an attorney or agent (acting in a representative capacity under 37 CFR 1.34(a)) upon the filing of a continuing application.

6. □ The decision by the Board of Patent Appeals and Interference rendered on _____ and because the period for seeking court review of the decision has expired and there are no allowed claims.

7. ☒ The reason(s) below:

   A confirmation for abandoment was made on 10/22/2003 with Michael De Angelo. Phone number: 760 - 864 - 9500.

SAFET METJAHIC
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Petitions to revive under 37 CFR 1.137(a) or (b), or requests to withdraw the holding of abandonment under 37 CFR 1.181, should be promptly filed to minimize any negative effects on patent term.

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
www.uspto.gov

Paper No. 9

Greg T. Sueoka
FENWICK & WEST LLP
Two Palo Alto Square
Palo Alto, CA 94306

In re Application of:
Michael De Angelo
Application No. 09/284,113
Filed: April 7, 1999
For: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

DECISION ON REQUEST TO WITHDRAW AS ATTORNEY OR AGENT

This is a decision on the Request to Withdraw from Representation filed January 24, 2003.

A grantable request to withdraw as attorney of record should indicate thereon the present mailing addresses of the attorney(s) who is/are withdrawing from the record and of the applicant. The request for withdrawal must be signed by every attorney seeking to withdraw or contain a clear indication that one attorney is signing on behalf of another/others. A request to withdraw will not be approved unless at least 30 (thirty) days would remain between the date of approval and the later of the expiration date of a time to file a response or the expiration date of the maximum time period which can be extended under 37 C.F.R. § 1.136(a). The effective date of withdrawal being the date of decision and not the date of request. See M.P.E.P. § 402.06. 37 C.F.R. § 1.36 further requires that the applicant or patent owner be notified of the withdrawal of the attorney or agent.

The request is **GRANTED.**

All future communications from the Office will be directed to the below-listed address until otherwise notified by applicant. This correspondence address is provided by the withdrawn attorney(s). Applicant is reminded of the obligation to promptly notify the Patent and Trademark Office (Office) of any change in correspondence address to ensure receipt of all communications from the Office.

Vincent N. Trans
Special Program Examiner
Technology Center 2100
Computer Architecture, Software, and
Information Security
703-305-9750

cc: Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

08-11-04

Attorney Docket No.: 17776-002US1

DAe
A

#10

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : DeAngelo, Michael
Serial No. : 09/284,113
Filed : April 7, 1999
Title : SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

Art Unit : 2171
Examiner : Cam N. Nguyen

MAIL STOP PETITIONS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION TO REVIVE APPLICATION UNDER 37 C.F.R. § 1.137(b)

Applicant hereby petitions under 37 C.F.R. § 1.137(b) to revive the above application for failure to respond to the non-final office action mailed February 11, 2003.

Enclosed is 1) a declaration of Michael de Angelo in support of this petition and upon which this petition is based, 2) a response to the non-final office action mailed February 11, 2003, to continue prosecution of the application, and 3) a check for $665 in payment of the petition fee by a small entity as set forth in 37 C.F.R. § 1.17(m).

Applicant submits that the entire delay in filing the required reply from the due date for the reply until the filing of a grantable petition pursuant to 37 C.F.R. § 1.137(b) was unintentional.

Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: Aug. 9, 2004

Tamara Fraizer
Reg. No. 51,699

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

50227801.doc

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 321 384 896 US

August 9, 2004
Date of Deposit

Attorney's Docket No.: 17776-002US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: DeAngelo, Michael	Art Unit	: 2171
Serial No.	: 09/284,113	Examiner	: Cam N. Nguyen
Filed	: April 7, 1999		
Title	: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF MICHAEL DE ANGELO IN SUPPORT OF PETITION TO REVIVE APPLICATION UNDER 37 C.F.R. § 1.137(b)

I, Michael De Angelo, declare and state as follows:

1. My citizenship, residence and address are as stated below by my name and signature.

2. I believe that I am the original, first and sole inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled: System And Method For Creating And Manipulating Information Containers With Dynamic Registers. My patent application is based upon over twenty-five years of committed work and development.

3. Prior to the Patent Office's issuance on February 11, 2003, of an Office Action pertaining to my application, I was informed by patent counsel that they were no longer representing me because I had not been able to pay their bills.

4. During the entire year of 2003 and until the present time, I have suffered severe hardship due to the criminal destruction of a corporation for which I was the Chairman and CEO. This matter was investigated by the FBI and the Department of Justice, and a Federal subpoena was issued to the perpetrators. The destruction of the corporation and multiple related lawsuits caused me extreme mental anguish and serious financial difficulties.

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date of Deposit

Signature

Typed or Printed Name of Person Signing Certificate

5. I had no understanding of the timeline for my response to the first office action issued with respect to my application and was unable to pursue the matter during this time.

6. The entire delay in filing of the required reply to the PTO Office Action of February 11, 2003, from the due date for the reply until the filing of the accompanying petition was unintentional.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Full name: MICHAEL DE ANGELO
Signature:
Date: August 5, 2004
Residence and Address: 3700 Andreas Hills Drive, Palm Springs, CA 92264
Citizen of: United States of America

Attorn Docket No.: 17776-002US1

#11/a

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	DeAngelo, Michael	Art Unit	:	2171
Serial No.	:	09/284,113	Examiner	:	Cam N. Nguyen
Filed	:	April 7, 1999			
Title	:	SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS			

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT IN REPLY TO ACTION OF FEBRUARY 11, 2003

Please amend the above-identified application as follows:

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 321 384 896 US

August 9, 2004
Date of Deposit

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

a1

1. (Currently Amended) An apparatus for transmitting, receiving and manipulating information on a computer system,~~.~~ the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element having information;

a plurality of registers, the plurality of registers forming part of the container and including~~,~~

a first register ~~of the plurality of registers~~ for storing a unique container identification value and~~,~~

a second register having a representation designating time and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus event time~~of the plurality of registers that stores information and evolves according to the relationship, use and interaction of the container with other containers, processes and systems~~; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems ~~and~~ or processes.

2. (Currently Amended) The apparatus of claim 1 or 37, wherein the information element is one from the group of text, graphic images, video, audio, a digital pattern, a process, a nested container, bit, natural number and a system.

3. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the plurality of registers includes at least one container history register for storing information

regarding past interaction of the container with other containers, systems or processes, the container history register being ~~modified~~modifiable.

a1

4. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the plurality of registers includes at least one system history register for storing information regarding past interaction of the container with different operating system and network processes.

5. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the plurality of registers includes at least one prede~~t~~fined register, the predefined register being a register associated with an editor for user selection and ~~, the predefined register~~ being appendable to any container.

6. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the plurality of registers includes a user-created register, the user-created register being generated by the user, ~~one or more of which is~~and being appendable to any container.

7. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the plurality of registers includes a system-defined -register, the system-defined register being set, controlled and used by the system, ~~one or more of which is~~ and being appendable to any container.

8. (Cancelled)

9. (Currently Amended) The ~~system~~ apparatus of claim ~~8~~1, wherein the plurality of registers includes:

an active time register for identifying times at which the container will act upon other containers, processes, systems or gateways;

a~~n~~ passive time register for identifying times at which the container can be acted upon by other containers, processes, systems~~,~~ or gateways; and

a neutral ~~tune~~ time register for identifying times at which the container may interact with other containers, processes, systems or gateways.

a1

10. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the plurality of registers includes at least one acquire register for controlling whether the container adds a register or a container from other containers when interacting with them.

11-13. (Cancelled)

14. (Currently Amended) The ~~system~~ apparatus of claim 37~~11~~,~~.~~ wherein the plurality of registers includes:

an active space register for identifying space in which the container will act upon other containers, processes, systems or gateways;

a~~n~~ passive space register for identifying ~~from~~ space in which the container can be acted upon by other containers, processes, systems~~,~~ or gateways; and

a neutral ~~time~~ space register for identifying space in which the container may interact with other containers, processes, systems or gateways.

15. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the gateway includes means for acting upon another container, the means for acting upon another container using the plurality of registers to determine whether and how the container acts upon other containers.

16. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the gateway includes means for allowing interaction, the means for allowing interaction using the plurality of registers to determine whether and how another container can act upon the container.

17. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the gateway includes means for gathering information, the means for gathering information recording register information from other containers, systems ~~and~~ or processes that interact with the container.

a1

18. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the gateway includes means for reporting information, the means for reporting information providing register information to other containers, systems ~~and~~ or processes that interact with the container.

19. (Currently Amended) The ~~system~~ apparatus of claim 1 or 37, wherein the gateway includes an expert system including rules defining the interaction of the container with other containers, systems ~~and~~ or processes.

20-36. (Cancelled)

37. (New) An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element having information;

a plurality of registers, the plurality of registers forming part of the container and including

a first register for storing a unique container identification value and

a second register having a representation designating space and governing interactions ofthe container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus three-dimensional space; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems or processes.

REMARKS

Claims 1-36 were pending in the present application. Claims 1-7, 9-10, 14-19 have been amended. Claims 8, 11-13, and 20-36 have been cancelled. Claim 37 has been added.

No new matter has been added by way of this amendment. Support for newly added claim 37 can be found in the specification, for example, at page 5 lines 20-26, page 19 lines 6-7, and page 35 lines 25-30. Reconsideration and reexamination are respectfully requested in view of the amendments and following remarks.

Claim Rejections 35 USC § 102(e)

The Examiner rejected claims 1-36 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,016,495 to McKeehan et al. ("McKeehan"). The applicant respectfully disagrees.

The applicant's invention as defined in the amended claims provides an apparatus for transmitting, receiving, and manipulating information on a computer system, and includes an information element, a plurality of registers, and a gateway. Claim 1 requires a gateway that controls the interactions of the container, and a register that has a representation of time that governs the container's interactions according to the utility of the container's information relative to external-to-the-apparatus time.

In contrast, McKeehan describes an object-oriented framework mechanism that provides an infrastructure for persistent storage. McKeehan does not describe or suggest, as required by amended claim 1, "a second register for controlling the relationship of the container with other containers, systems or processes, the second register having a representation designating time and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus event time".

Rather, McKeehan describes a "LockManager class ... responsible for providing concurrency control of objects stored in persistent containers" (Column 19, lines 25-27). Applicant respectfully submits that McKeehan's LockManager class does not "govern[] interactions ... according to utility of information ... relative to an external-to-the-apparatus event time", as required by claim 1, because the LockManager class controls simultaneous access to an object by multiple objects without regard to the utility of such access relative to "an external-to-the-apparatus event time."

Applicant's new claim 37 requires a gateway that controls the interactions of the container, and a register that has a representation of space that governs the container's interactions according to the utility of the container's information relative to external-to-the-apparatus space. McKeehan does not describe or suggest, as required by claim 37, "a second register for controlling the relationship of the container with other containers, systems and processes, the second register having a representation designating space and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus three-dimensional space".

As noted by the Examiner with respect to previously asserted claim 11, McKeehan describes a "[c]omputer system 800 [that] utilizes well known virtual addressing mechanisms that allow the programs of computer system 800 to behave as if they only have access to a large, single storage entity (referred to herein as computer system memory) instead of access to multiple, smaller storage entities such as main memory ..." (Column 15, line 66 to column 16, line 5). Applicant respectfully submits that McKeehan's virtual addressing mechanism is limited to the memory of the computer system, and as such, does not suggest "governing interactions ... according to utility of information ... relative to an external-to-the-apparatus three-dimensional space".

Accordingly, the applicant respectfully submits that claim 1, as amended, and new claim 37 are allowable. Claims 2-7, 9-10, and 14-19 depend from claim 1 or 37, and are allowable for at least the reasons discussed for claims 1 or 37. Allowance of claims 1-7, 9-10, 14-19 and 37 is thereby respectfully requested.

Submitted herewith is a petition for revival of the application and associated fees. Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: Aug. 9, 2004

Tamara Fraizer
Reg. No. 51,699

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

Attorney's Docket No.: 17776-002US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	DeAngelo, Michael	Art Unit	:	2171
Serial No.	:	09/284,113	Examiner	:	Cam N. Nguyen
Filed	:	April 7, 1999			
Title	:	SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS			

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER

Correspondence relating to this application is enclosed. The required fees are computed below. Please apply any charges not covered, or any credits, to Deposit Account No. 06-1050.

Total Claims	28	-	36	=	0	$0
Independent	2	-	3	=	0	$0
First Presentation of Multiple Dependent Claims						$145
Applicant hereby petitions under 37 C.F.R. §1.136 for a 0 month extension of time.						$0
TOTAL FEE DUE						$145

A check for $145 is attached.

Respectfully submitted,

Date: Aug. 9, 2004

Tamara Fraizer
Reg. No. 51,699

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

50231087.doc

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 321 388 535 US

March 8, 2004
Date of Deposit

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov
DIW 09-04

Paper No. 12

MICHAEL DE ANGELO
INFORMATION EQUITY CORPORATION
100 SOUTH SUNRISE BOULEVARD, SUITE 470
PALM SPRINGS CA 92262

In re Application of :
Michael DeAngelo :
Application No. 09/284,113 : ON PETITION
Filed: 7 April, 1999 :
Att'y Docket No. 3726-US :

This is a decision on the petition under 1.137(b),[1] filed on 10 August, 2004, to revive the above-identified application.

The petition is **GRANTED**.

This application became abandoned on 12 May, 2003, for failure to timely submit a response to the non-final Office action mailed on 11 February, 2003, which set a three (3) month shortened statutory period for reply. No extensions of the time for reply

[1] Effective December 1, 1997, the provisions of 37 CFR 1.137(b) now provide that where the delay in reply was unintentional, a petition may be filed to revive an abandoned application or a lapsed patent pursuant to 37 CFR 1.137(b). A grantable petition filed under the provisions of 37 CFR 1.137(b) must be accompanied by:

(1) the required reply, unless previously filed. In a nonprovisional application abandoned for failure to prosecute, the required reply may be met by the filing of a continuing application. In a nonprovisional application filed on or after June 8, 1995, and abandoned for failure to prosecute, the required reply may also be met by the filing of a request for continued examination in compliance with § 1.114. In an application or patent, abandoned or lapsed for failure to pay the issue fee or any portion thereof, the required reply must be the payment of the issue fee or any outstanding balance thereof. In an application abandoned for failure to pay the publication fee, the required reply must include payment of the publication fee.

(2) the petition fee as set forth in 37 CFR 1.17(m);

(3) a statement that the entire delay in filing the required reply from the due date for the reply until the filing of a grantable petition pursuant to 37 CFR 1.137(b) was unintentional. The Commissioner may required additional information where there is a question whether the delay was unintentional; and

(4) any terminal disclaimer (and fee as set forth in 37 CFR 1.20(d)) required pursuant to 37 CFR 1.137(c)).

in accordance with 37 CFR 1.136(a) were obtained. Notice of Abandonment was mailed on 30 October, 2003.

There is no indication that petition herein was ever empowered to prosecute the instant application. If petitioner desires to receive future correspondence regarding this application, the appropriate power of attorney documentation must be mailed. A courtesy copy of this decision will be mailed to petitioner. However, all future correspondence will be directed to the address of record until such time as appropriate instructions are received to the contrary.

The application file is being forwarded to Technology Center 2100 for further processing.

Telephone inquiries concerning this matter may be directed to the undersigned at (703)308-6918.

D Wood

Douglas I. Wood
Senior Petitions Attorney
Office of Petitions

cc: Fish & Richardson, P.C.
500 Arguello Street, Suite 500
Redwood City, CA 94063

L Number	Hits	Search Text	DB	Time stamp
1	6057	(plural43 or multiple) near2 register$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:46
2	112782	register$3 near5 (time or lock$3 or clock$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:47
3	50244	register$3 near3 (time)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:14
4	1681	((plural43 or multiple) near2 register$3) and (register$3 near3 (time))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:14
5	947	(((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:15
6	45	((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (707/$ or 709/$).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:37
7	414	interaction$3 near3 (databases or storages or containers)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:38
8	0	((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (interaction$3 near3 (databases or storages or containers))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:37
9	1399	interaction$3 with (databases or storages or containers)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:48
10	0	((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (interaction$3 with (databases or storages or containers))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:38
11	66	(register$3 near5 (time or lock$3 or clock$3)) and (interaction$3 with (databases or storages or containers))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:39
12	66	((register$3 near5 (time or lock$3 or clock$3)) and (interaction$3 with (databases or storages or containers))) not (((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (707/$ or 709/$).ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:39
13	23	(((register$3 near5 (time or lock$3 or clock$3)) and (interaction$3 with (databases or storages or containers))) not (((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (707/$ or 709/$).ccls.)) and @AD<19980130	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:40

14	6991	(plural43 or multiple) near2 (register$3 or registration or registrative)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:18
15	81602	(register$3 or registration or registrative) near5 time	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:08
16	1389	((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:47
17	0	(interaction$3 with (databases or storages or containers)) and (((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:47
18	31	(register$3 or registration or registrative) with interaction$3 with (databases or storages or containers)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:48
19	6	((register$3 or registration or registrative) with interaction$3 with (databases or storages or containers)) and @AD<19980130	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:49
20	98100	(register$3 or registration or registrative).ti.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:51
21	262	(((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative).ti.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:51
22	12321	((register$3 or registration or registrative) with time).clm.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:51
23	47	((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative).ti.)) and (((register$3 or registration or registrative) with time).clm.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:52
24	47	(((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative).ti.)) and (((register$3 or registration or registrative) with time).clm.)) not (((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (707/$ or 709/$).ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 08:52

25	57076	(register$3 or registration or registrative) near3 (time or time$stamp or timestamp)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:14
26	1050	(((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative) near3 (time or time$stamp or timestamp))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:10
27	203	((register$3 or registration or registrative).ti.) and ((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative) near3 (time or time$stamp or timestamp)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:12
29	0	((((register$3 or registration or registrative) with time).clm.) and (((register$3 or registration or registrative).ti.) and ((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative) near3 (time or time$stamp or timestamp))))) not ((((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative).ti.)) and (((register$3 or registration or registrative) with time).clm.)) not (((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (707/$ or 709/$).ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:12
28	43	(((register$3 or registration or registrative) with time).clm.) and (((register$3 or registration or registrative).ti.) and ((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative) near3 (time or time$stamp or timestamp))))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:11
30	4	(((register$3 or registration or registrative).ti.) and ((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative) near3 (time or time$stamp or timestamp)))) and (707/$ or 709/$).ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:12
31	222	(register$3 or registration or registrative) near3 (time$stamp or timestamp)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:18

32	0	((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near3 (time$stamp or timestamp)) and @AD<19980130	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:15
33	68	((register$3 or registration or registrative) near3 (time$stamp or timestamp)) and @AD<19980130	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:15
34	68	(((register$3 or registration or registrative) near3 (time$stamp or timestamp)) and @AD<19980130) not ((((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative).ti.)) and (((register$3 or registration or registrative) with time).clm.)) not (((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (707/$ or 709/$).ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:15
35	11	((((register$3 or registration or registrative) near3 (time$stamp or timestamp)) and @AD<19980130) not ((((((plural43 or multiple) near2 (register$3 or registration or registrative)) and ((register$3 or registration or registrative) near5 time) and @AD<19980130) and ((register$3 or registration or registrative).ti.)) and (((register$3 or registration or registrative) with time).clm.)) not (((((plural43 or multiple) near2 register$3) and (register$3 near3 (time))) and @AD<19980130) and (707/$ or 709/$).ccls.))) and interactions	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:15
36	8	((register$3 or registration or registrative) near3 (time$stamp or timestamp)).ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:18
37	8	((plural43 or multiple) near2 (register$3 or registration or registrative)) same (timestamp$ or time$stamp$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:19
38	1292	((plural43 or multiple) near2 (register$3 or registration or registrative)) same (time or timestamp$ or time$stamp$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:20
39	704	(((plural43 or multiple) near2 (register$3 or registration or registrative)) same (time or timestamp$ or time$stamp$3)) and @AD<19980130	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:20
40	325	((((plural43 or multiple) near2 (register$3 or registration or registrative)) same (time or timestamp$ or time$stamp$3)) and @AD<19980130) and ((control or govern$3) with (interaction$3 or operation))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/10/26 09:22

41	16	(((((plural43 or multiple) near2 (register$3 or registration or registrative)) same (time or timestamp$ or time$stamp$3)) and @AD<19980130) and ((control or govern$3) with (interaction$3 or operation))) and (707/$ or 709/$ or 705/$).ccls.	USPAT;
			US-PGPUB;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				2004/10/26 09:24

8

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 11/02/2004

Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

EXAMINER	
NGUYEN, CAM LINH T	
ART UNIT	PAPER NUMBER
2161	

DATE MAILED: 11/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/284,113	**Applicant(s)** DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	**Art Unit** 2161	

80

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 August 2004*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-7,9,10,14-19 and 37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7,9,10,14-19 and 37* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) □ Notice of Informal Patent Application (PTO-152)
6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's amendments to claims 1 – 37 are acknowledged. Consequently, claims 8, 11 – 13, 20 – 36 are cancelled. Claims 1 – 7, 9 – 10, 14 – 19, and 37 are currently pending.

Specification

1. This application does not contain an abstract of the disclosure as required by 37 CFR 1.72(b). An abstract on a separate sheet is required.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1 – 7, 10, 15 – 19, and 37 are rejected under 35 U.S.C. 102(e) as being anticipated by Chiussi et al (U.S. 6,075,791).

♦ As per claim 1, 37,

Chiussi discloses an apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

- " An information element having information" See Fig. 1, element 2, col.4, lines 10 - 13.
- " A plurality of registers (Fig. 1 - 3), the plurality of registers forming part of the container and including:

- "A first register for storing a unique container identification value" Fig. 3, element 30-1, col. 5, lines 1 – 2.
- " A second register" See Fig. 3, element 50 – 1, col. 5, lines 4 - 5.

- " A gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, system or processes" See Fig. 1, element 1, and Fig. 2, col. 4, lines 10 - 39.

♦ As per claim 2, Chiussi discloses:

- "The information element is one from the group of text, graphic…a system" col.4, lines 10 – 13 of Chiussi.

♦ As per claim 3 – 4, Chiussi discloses:

- "One container history register for storing information regarding past interaction of the container with other container… modified" See col. 6, lines 46 – 63, of Chiussi.

♦ As per claim 5 – 7, Chiussi discloses:

- "Plurality of registers include at least one predefined register" and "Plurality of registers include a user created register" See col. 17, lines 22.

♦ As per claims 9, 14,

- " An active time register for identifying times" See col. 5, lines 20 – 23 of Chiussi.

♦ As per claim 10, Chiussi discloses:

- "Plurality of registers include at least one acquire register" See col. 6, lines 34 – 41 of Chiussi

♦ As per claim 15 - 19, Chiussi discloses:

Because the server controls the operation of registers, it must including the means of "allowing interaction, gather information, reporting information, and including the rules defining the interaction of the container".

Conclusion

1. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

2. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Ramkumar et al (U.S. 6,173,280) discloses a method and apparatus for generating weighted association rules.
- Kawaguchi et al (U.S. 6,154,782) discloses a server switching between communication modes for clients coupled to the server.

- Chang et al (U.S. 6,198,738) discloses a communications between the public switched telephone network and packetized data networks.
- Teper et al (U.S. 5,815,665) discloses a system and method for providing trusted brokering services over a distributed network.

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CamLinh Nguyen whose telephone number is (571) 272-4024. The examiner can normally be reached on Monday - Friday from 8:00 am to 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Safet Metjahic, can be reached on (571) 272- 4023. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 305-3900.

Cam-Linh Nguyen
Art Unit 2161

LN

SAFET METJAHIC
SORY PATENT EXAMINER
LOGY CENTER 2100

Notice of References Cited	Application/Control No. 09/284,113	Applicant(s)/Patent Under Reexamination DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	Art Unit 2161	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,173,280 B1	01-2001	Ramkumar et al.	707/6
	B	US-6,154,782 A	11-2000	Kawaguchi et al.	709/239
	C	US-6,198,738 B1	03-2001	Chang et al.	370/352
	D	US-5,815,665 A	09-1998	Teper et al.	709/229
	E	US-6,075,791 A	06-2000	Chiussi et al.	370/412
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIBDATASHEET

Bib Data Sheet

CONFIRMATION NO. 1910

SERIAL NUMBER 09/284,113	FILING DATE 04/07/1999 RULE	CLASS 707	GROUP ART UNIT 2161	ATTORNEY DOCKET NO. 3726-US

APPLICANTS

MICHAEL DE ANGELO, SANTA BARBARA, CA;

** CONTINUING DATA ************************** Yes

This application is a 371 of PCT/US99/01988 01/28/1999 LN
which claims benefit of 60/073,209 01/30/1998

** FOREIGN APPLICATIONS ******************** None LN

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 04/12/2000

Foreign Priority claimed □ yes ☑ no 35 USC 119 (a-d) conditions met □ yes ☑ no □ Met after Allowance Verified and Acknowledged NguyenCauLinh Examiner's Signature LN Initials	STATE OR COUNTRY CA	SHEETS DRAWING 30	TOTAL CLAIMS ~~26~~ 16	INDEPENDENT CLAIMS 2

ADDRESS
Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs , CA
92262

TITLE
SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

FILING FEE RECEIVED 669	FEES: Authority has been given in Paper No. _________ to charge/credit DEPOSIT ACCOUNT No. _________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other ____________ □ Credit

Index of Claims

Application No. 09/284,113	Applicant(s) DE ANGELO, MICHAEL	
Examiner CamLinh Nguyen	Art Unit 2161	

√	Rejected	—	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date								
Final	Original	10/26/04								
	1	v								
	2	v								
	3	v								
	4	v								
	5	v								
	6	v								
	7	v								
	8									
	9	v								
	10	v								
	11									
	12									
	13									
	14	v								
	15	v								
	16	v								
	17	v								
	18	v								
	19	v								
	20									
	21									
	22									
	23									
	24									
	25									
	26									
	27									
	28									
	29									
	30									
	31									
	32									
	33									
	34									
	35									
	36									
	37	v								
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	50									

Claim		Date								
Final	Original									
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									

Claim		Date								
Final	Original									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

Search Notes	Application No. 09/284,113	Applicant(s) DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	Art Unit 2161	

SEARCHED

Class	Subclass	Date	Examiner
707	6	10/25/2004	LINH

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST search	10/25/2004	LINH

Attorney's Docket No.: 17776-002US1

OFFICIAL COMMUNICATION FACSIMILE:

OFFICIAL FAX NO: (703) 872-9306

Number of pages including this page 3 pages

Applicant : DeAngelo, Michael
Serial No. : 09/284,113
Filed : April 7, 1999

Art Unit : 2171
Examiner : Cam N. Nguyen

Title : System and Method for Creating and Manipulating Information Containers with Dynamic Registers

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attached to this facsimile communication cover sheet is Power of Attorney by Assignee and Election of Assignee to Conduct Prosecution to Exclusion of Inventors, faxed this 3rd day of November, 2004, to the United States Patent and Trademark Office.

Respectfully submitted,

Date: November 3, 2004

Tamara Fraizer
Reg. No. 51,699

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Fax: (650) 839-5071

50245504.doc

NOTE: This facsimile is intended for the addressee only and may contain privileged or confidential information. If you have received this facsimile in error, please immediately call us collect at (650) 839-5070 to arrange for its return. Thank you.

Applicant : DeAngelo, Michael
Serial No. : 09/284,113
Filed : April 7, 1999
Page : 2 of 2

Attorney's Docket No.: 17776-002US1

Tamara Fraizer
FISH & RICHARDSON P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

Signature: Michael De Angelo

Typed name: Michael DeAngelo

Title:

Assignee: EMATRIX Corporation or Pattern Intelligence, Inc

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

50240573.doc

Attorney's Docket No.: 17776-002US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: DeAngelo, Michael	Art Unit	: 2171
Serial No.	: 09/284,113	Examiner	: Cam N. Nguyen
Filed	: April 7, 1999		
Title	: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

POWER OF ATTORNEY BY ASSIGNEE AND ELECTION OF ASSIGNEE TO CONDUCT PROSECUTION TO EXCLUSION OF INVENTORS

The undersigned, as authorized representative of the assignee of the entire right, title and interest in the above-identified application, hereby appoints

Subroto Bose, Reg. No. 55,014	Tim H. Pham, Reg. No. 48,589
David J. Goren, Reg. No. 34,609	Hans R. Troesch, Reg. No. 36,950
Brian J. Gustafson, Reg. No. Reg. No. 52,978	Kelvin Vivian, Reg. No. 53,727
Tamara Fraizer, Reg. No. 51,699	Elissa Wang, Reg. No. 48,668
Mark D. Kirkland, Reg. No. 40,048	Jennifer Zanocco, Reg. No. 54,563

as its attorneys to prosecute the application and to transact all business in the Patent and Trademark Office connected therewith with full powers of substitution and revocation, said appointment to be to the exclusion of the inventors and their attorney(s) in accordance with the provisions of 37 CFR §3.71 *et seq.* of the Patent Office Rules of Practice.

Ownership is in the assignee by virtue of the assignment documents filed on April 7, 1999. The documents evidencing ownership have been reviewed and to the best of the assignee's knowledge and belief, title is in the assignee.

Please direct all communications regarding the application to the attorney at the address and telephone numbers indicated below.

I hereby certify that this correspondence is being transmitted by facsimile to the Patent and Trademark Office on the date indicated below.

____
Date of Transmission

____
Signature

____

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 11/02/2004

Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

EXAMINER	
NGUYEN, CAM LINH T	
ART UNIT	PAPER NUMBER
2161	

DATE MAILED: 11/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	09/284,113	DE ANGELO, MICHAEL	
	Examiner	Art Unit	
	CamLinh Nguyen	2161	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 August 2004*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-7,9,10,14-19 and 37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7,9,10,14-19 and 37* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's amendments to claims 1 – 37 are acknowledged. Consequently, claims 8, 11 – 13, 20 – 36 are cancelled. Claims 1 – 7, 9 – 10, 14 – 19, and 37 are currently pending.

Specification

1. This application does not contain an abstract of the disclosure as required by 37 CFR 1.72(b). An abstract on a separate sheet is required.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1 – 7, 10, 15 – 19, and 37 are rejected under 35 U.S.C. 102(e) as being anticipated by Chiussi et al (U.S. 6,075,791).

◊ As per claim 1, 37,

Chiussi discloses an apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

- " An information element having information" See Fig. 1, element 2, col.4, lines 10 - 13.
- " A plurality of registers (Fig. 1 - 3), the plurality of registers forming part of the container and including:

- "A first register for storing a unique container identification value" Fig. 3, element 30-1, col. 5, lines 1 – 2.
- " A second register" See Fig. 3, element 50 – 1, col. 5, lines 4 - 5.

- " A gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, system or processes" See Fig. 1, element 1, and Fig. 2, col. 4, lines 10 - 39.

♦ As per claim 2, Chiussi discloses:

- "The information element is one from the group of text, graphic…a system" col.4, lines 10 – 13 of Chiussi.

♦ As per claim 3 – 4, Chiussi discloses:

- "One container history register for storing information regarding past interaction of the container with other container… modified" See col. 6, lines 46 – 63, of Chiussi.

♦ As per claim 5 – 7, Chiussi discloses:

- "Plurality of registers include at least one predefined register" and "Plurality of registers include a user created register" See col. 17, lines 22.

♦ As per claims 9, 14,

- " An active time register for identifying times" See col. 5, lines 20 – 23 of Chiussi.

♦ As per claim 10, Chiussi discloses:

- "Plurality of registers include at least one acquire register" See col. 6, lines 34 – 41 of Chiussi

♦ As per claim 15 - 19, Chiussi discloses:

Because the server controls the operation of registers, it must including the means of "allowing interaction, gather information, reporting information, and including the rules defining the interaction of the container".

Conclusion

1. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

2. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Ramkumar et al (U.S. 6,173,280) discloses a method and apparatus for generating weighted association rules.
- Kawaguchi et al (U.S. 6,154,782) discloses a server switching between communication modes for clients coupled to the server.

- Chang et al (U.S. 6,198,738) discloses a communications between the public switched telephone network and packetized data networks.
- Teper et al (U.S. 5,815,665) discloses a system and method for providing trusted brokering services over a distributed network.

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CamLinh Nguyen whose telephone number is (571) 272-4024. The examiner can normally be reached on Monday - Friday from 8:00 am to 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Safet Metjahic, can be reached on (571) 272- 4023. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 305-3900.

Cam-Linh Nguyen
Art Unit 2161

LN

SAFET METJAHIC
SORY PATENT EXAMINER
LOGY CENTER 2100

Notice of References Cited	Application/Control No. 09/284,113	Applicant(s)/Patent Under Reexamination DE ANGELO, MICHAEL
	Examiner CamLinh Nguyen	Art Unit 2161	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,173,280 B1	01-2001	Ramkumar et al.	707/6
	B	US-6,154,782 A	11-2000	Kawaguchi et al.	709/239
	C	US-6,198,738 B1	03-2001	Chang et al.	370/352
	D	US-5,815,665 A	09-1998	Teper et al.	709/229
	E	US-6,075,791 A	06-2000	Chiussi et al.	370/412
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	42605	(plural$3 or multiple) near3 register$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:18
L2	28564	1 and @AD<"19990407"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 09:50
L3	16264	active with register$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 09:50
L4	688	passive with register$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 09:51
L5	649	neutral with register$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 09:51
L6	0	2 and 3 and 4 and 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 09:51

L7	2824	2 and 3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 09:51
L8	25	2 and 3 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:16
L9	52525	register$3 with (ID or IDs or identif$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:17
L10	5583	2 and 9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:17
L11	117	10 and "707"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:17
L12	6831	(register$3 with (ID or IDs or identif$6)).clm.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:17

L13	17321	((plural$3 or multiple) near3 register$3).clm.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:18
L14	1032	12 and 13 and @AD<"19980130"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:18
L15	19	14 and "707"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:22
L16	3	14 and 3 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:23
L17	0	14 and 3 and 4 and 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/30 10:23

Attorney's Docket No.: 17776-002US1

OFFICIAL COMMUNICATION FACSIMILE:

OFFICIAL FAX NO: (703) 872-9306

Number of pages Including this page 15

Applicant : DeAngelo, Michael		Art Unit : 2171	
Serial No. : 09/284,113		Examiner : Cam N. Nguyen	
Filed : April 7, 1999			

Title : System and Method for Creating and Manipulating Information Containers with Dynamic Registers

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attached to this facsimile communication cover sheet is an AMENDMENT IN REPLY TO ACTION OF NOVEMBER 2, 2004, faxed this 3rd day of Janurary, 2005, to the United States Patent and Trademark Office.

Respectfully submitted,

Date: Janurary 3, 2004

Dorian Cartwright

Dorian Cartwright
Reg. No. 53,853

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Fax: (650) 839-5071

50254229.doc

NOTE: This facsimile is intended for the addressee only and may contain privileged or confidential information. If you have received this facsimile in error, please immediately call us collect at (650) 839-5070 to arrange for its return. Thank you.

Attorney's Docket No.: 17776-002US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant :	DeAngelo, Michael	Art Unit :	2171
Serial No. :	09/284,113	Examiner :	Cam N. Nguyen
Filed :	April 7, 1999		
Title :	SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS		

MAIL STOP AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INTERVIEW SUMMARY AND AMENDMENT
IN REPLY TO FINAL OFFICE ACTION OF NOVEMBER 2, 2004

Please amend the above-identified application as follows:

I hereby certify that this correspondence is being transmitted by facsimile to the Patent and Trademark Office on the date indicated below.

1/3/05
Date of Transmission

Dorian Cartwright
Signature

Dorian Cartwright

Amendments to the Specification:

Please delete previous abstract at page 50 and add the following new abstract, a clean version of which is also attached on a separate sheet:

--An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising an information element, a plurality of registers, and a gateway. The plurality of registers, form part of the container, and include a first register for storing a unique container identification value; a second register having a representation designating time and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus event time; an active time register for identifying times at which the container will act upon other containers, processes, systems or gateways; a passive time register for identifying times at which the container can be acted upon by other containers, processes, systems or gateways; and a neutral time register for identifying times at which the container may interact with other containers, processes, systems or gateways. Additional registers designate space for container interactions.--

Amendments to the Claims:

Claims 1, 10 and 37 are amended. Claims 9 and 14 are cancelled. Claims 38 and 39 are added. This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element having information;

a plurality of registers, the plurality of registers forming part of the container and including

a first register for storing a unique container identification value, ~~and~~

a second register having a representation designating time and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus event time,

an active time register for identifying times at which the container will act upon other containers, processes, systems or gateways,

a passive time register for identifying times at which the container can be acted upon by other containers, processes, systems or gateways, and

a neutral time register for identifying times at which the container may interact with other containers, processes, systems or gateways; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems or processes.

2. (Previously presented) The apparatus of claim 1 or 37, wherein the information element is one from the group of text, graphic images, video, audio, a digital pattern, a process, a nested container, bit, natural number and a system.

3. (Previously presented) The apparatus of claim 1 or 37, wherein the plurality of registers includes at least one container history register for storing information regarding past interaction of the container with other containers, systems or processes, the container history register being modifiable.

4. (Previously presented) The apparatus of claim 1 or 37, wherein the plurality of registers includes at least one system history register for storing information regarding past interaction of the container with different operating system and network processes.

5. (Previously presented) The apparatus of claim 1 or 37, wherein the plurality of registers includes at least one predefined register, the predefined register being a register associated with an editor for user selection and being appendable to any container.

6. (Previously presented) The apparatus of claim 1 or 37, wherein the plurality of registers includes a user-created register, the user-created register being generated by the user, and being appendable to any container.

7. (Previously presented) The apparatus of claim 1 or 37, wherein the plurality of registers includes a system-defined register, the system-defined register being set, controlled and used by the system, and being appendable to any container.

8. (Cancelled)

9. (Cancelled)

10. (Currently amended) The apparatus of claim 1 or 37, wherein the plurality of registers includes at least one acquire register for controlling whether the container adds a register ~~or a container~~ from other containers or adds a container from other containers when interacting with them.

11-13. (Cancelled)

14. (Cancelled)

15. (Previously presented) The apparatus of claim 1 or 37, wherein the gateway includes means for acting upon another container, the means for acting upon another container using the plurality of registers to determine whether and how the container acts upon other containers.

16. (Previously presented) The apparatus of claim 1 or 37, wherein the gateway includes means for allowing interaction, the means for allowing interaction using the plurality of registers to determine whether and how another container can act upon the container.

17. (Previously presented) The apparatus of claim 1 or 37, wherein the gateway includes means for gathering information, the means for gathering information recording register information from other containers, systems or processes that interact with the container.

18. (Previously presented) The apparatus of claim 1 or 37, wherein the gateway includes means for reporting information, the means for reporting information providing register information to other containers, systems or processes that interact with the container.

19. (Previously presented) The apparatus of claim 1 or 37, wherein the gateway includes an expert system including rules defining the interaction of the container with other containers, systems or processes.

20-36. (Cancelled)

37. (Currently amended) An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element having information;

a plurality of registers, the plurality of registers forming part of the container and including

a first register for storing a unique container identification value, ~~and~~

a second register having a representation designating space and governing interactions ~~ofthe~~ of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus three-dimensional space,

an active space register for identifying space in which the container will act upon other containers, processes, systems or gateways,

a passive space register for identifying space in which the container can be acted upon by other containers, processes, systems or gateways,

a neutral space register for identifying space in which the container may interact with other containers, processes, systems or gateways; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems or processes.

38. (New) An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element having information;

a plurality of registers, the plurality of registers forming part of the container and including

a first register for storing a unique container identification value,

a second register having a representation designating time and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus event time, and

at least one acquire register for controlling whether the container adds a register from other containers or adds a container from other containers when interacting with them; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems or processes.

39. (New) An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising:

an information element having information;

a plurality of registers, the plurality of registers forming part of the container and including

a first register for storing a unique container identification value, a second register having a representation designating time and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus event time, and

at least one acquire register for controlling whether the container adds a register from other containers or adds a container from other containers when interacting with them; and

a gateway attached to and forming part of the container, the gateway controlling the interaction of the container with other containers, systems or processes.

REMARKS

Claims 1-7, 9, 10, 14-19, and 37 were presented for examination. In a final office action mailed November 2, 2004, Examiner objected to the specification, and rejected claims 1-7, 9, 10, 14-19, and 37 under 35 U.S.C. § 102(e). On January 29, 2004, Examiner granted a telephone interview with Applicant's attorney.

In response, the specification is amended. Claims 1, 10 and 37 are amended. Claims 38 and 39 are added without introducing any new matter. Claims 9 and 14 are cancelled. Applicant thanks Examiner for examination and the subsequent interview, and now requests reconsideration of claims 1-7, 10, 15-19, and 37-39 in light of the following remarks.

I. Summary of the Interview

During the interview, Applicant argued that Chiussi fails to disclose containers configured for interactions with other containers as recited in independent claim 1 as an example. Furthermore, Applicant argued that Chiussi fails to disclose an active time register, a passive time register and/or a neutral time register as disclosed in dependent claim 9. While failing to reach agreement with respect to claim 1, Examiner agreed to reconsider claim 9 and other claims based on a formal communication.

II. Objection to the Specification

The specification was objected to because, according to the final action, the application did not contain an abstract of the disclosure. Applicant has amended the specification to include a new abstract. Applicant has also included the abstract a separate sheet. Therefore, Applicant respectfully submits that the specification should no longer be objected to.

III. Rejections under § 102(e)

Claims 1-7, 9, 10, 14-19, and 37 were rejected under § 102(e) as being anticipated by U.S. Patent No. 6,075,791 issued to Chiussi et al. ("Chiussi"). Applicant respectfully traverses the rejections as follows.

A. Claim 1

Claim 1 has been amended to include the limitations of claim 9, and is directed to an apparatus including a plurality of containers. Claim 1 further recites that each container comprises an active time register, a passive time register, and a neutral time register. Since the limitations are identical to a previously submitted claim, Applicant submits that no further searching is necessary.

Chiussi discloses a server 100 which services a plurality of queues having guaranteed data transfer rates and data transfer delays. (Abstract). A queue contains a connection identifier register 30-i, and a time stamp register 50-i. (col. 4, ln. 67-col. 5, ln. 5). The server 100 generates a new timestamp when a new packet reaches the head of a queue. (Col. 5, ll. 20-22)

However, Chiussi does not teach or suggest limitations recited in claim 1. Specifically, while the "active time register" of claim 1 identifies "times at which the container will act" (i.e., "upon other containers, processes, systems, or gateways"), the queue of Chiussi merely logs a time that a new packet reached the head of queue. Similarly, the "passive time register" of claim 1 identifies "times at which the container can be acted upon", and the "neutral time register" identifies "times at which the container may interact," neither of which are disclosed by Chiussi. Moreover, while claim 1 recites containers having a "second register...governing interactions with other containers," the queues of Chiussi are unable to interact with other queues because they have no awareness of other queues. Nor does Chiussi disclose any specific time parameters for such interaction. Thus, Chiussi fails to disclose the active time register, the passive time register, the neutral time register, or the second register as recited in claim 1.

Because Chiussi does not disclose every limitation of claim 1, Applicant respectfully submits that claim 1 is patentable over Chiussi.

B. Claim 37

Claim 37 has been amended to include the limitations of claim 14, and is directed to an apparatus including a plurality of containers. Claim 37 further recites that each container comprises an active space register, a passive space register, and a neutral space register. Since the limitations are identical to a previously submitted claim, Applicant submits that no further searching is necessary.

Examiner relies on the same disclosure for claim 37 as discussed above with respect to claim 1.

However, Chiussi does not teach or suggest limitations recited in claim 37. Specifically, the cited portions of Chiussi, in disclosing merely a connection identification register and a timestamp register, do not disclose a " second register designating space and governing interactions of the container with other containers" as recited in claim 37. Similarly, Chiussi does not specifically disclose an "active space register" to identify a "space in which the container will act upon, a "passive space register" to identify a "space in which the container can be acted upon," nor the "neutral space register" to identify a "space in which the container may interact." Also, while claim 37 recites a "second register...governing interactions of the container with other containers," the queues of Chiussi are unable to interact with other queues as discussed. Thus, Chiussi fails to disclose the active space register, the space time register, the neutral space register, or the second register as recited in claim 37.

Because Chiussi does not disclose every limitation of claim 37,, Applicant respectfully submits that claim 37 is patentable over Chiussi.

C. Claim 38

New claim 38 includes the limitations of previously presented claims 1 and 10, and is directed to an apparatus including a plurality of containers. Claim 38 further recites that each container comprises at least one acquire register. Because the limitations are identical to a previously submitted claim, Applicant submits that no further searching is necessary.

During the course of the interview, Examiner indicated that Chiussi discloses a server 100 that increments the content of register 115, that increments the content of queue length register 60-i, that mathematically or logically adds the content of rate register 40-i to the content of register 110, and that adds the product of the contents of timestamp register 50-i and rate register 40-i to the content of register 123. (Col. 8, ll. 3-24).

However, Chiussi does not teach or suggest limitations recited in claim 38 Specifically, the containers of claim 38 include an "acquire register" that can "control[] whether the container adds a register from other containers or adds a container from other containers." Thus, the container of claim 38 can acquire the register itself from another container rather than merely mathematically adding (or logically adding) the contents of two registers together as disclosed in Chiussi. Moreover, whereas the container of claim 38 can condition "whether" a register or container is added to the container the queues of Chiussi present no conditions since the centrally-controlled queues are unaware that other queues exist . Nor does the server of Chiussi assist in the acquisition of a register of one queue by another queue. As such, the queues of Chiussi are not able add registers directly from other queues. Nor are the queues able to add another queue. Moreover, the purpose of Chiussi, for sharing communication access between queues as designated by the server, would be foreclosed by adding one queue to another queue. Thus, Chiussi fails to disclose the acquire register as recited in claim 38.

Because Chiussi does not disclose every limitation of claim 38, Applicant respectfully submits that claim 38 is patentable over Chiussi.

D. Claim 39

New claim 39 includes the limitations of previously presented claims 37 and 10, and is directed to an apparatus including a plurality of containers. Claim 39 further recites that each container comprises at least one acquire register. Since the limitations are identical to a previously submitted claim, Applicant submits that no further searching is necessary.

Examiner relies on the same disclosure for claim 39 as discussed above with respect to claim 1.

However, Chiussi does not teach or suggest limitations recited in claim 39 for at least the same reasons as discussed above with respect to claim 39. Also, as discussed with respect to claim 37, Chiussi fails to disclose a register designating space. Thus, Chiussi fails to disclose the acquire register as recited in claim 38.

Because Chiussi does not disclose every limitation of claim 39, Applicant respectfully submits that claim 39 is patentable over Chiussi.

E. Dependent Claims

Because claims 2-7, 10, and 15-19 depend from patentable base claims, these claims are patentable for at least the same reasons.

Please apply $200 for excess claim fees, and any other charges or credits, to deposit account 06-1050.

Respectfully submitted,

Date: 1/3/05

Dorian Cartwright

Dorian Cartwright
Reg. No. 53,583

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

50249246.doc

ABSTRACT OF THE DISCLOSURE

An apparatus for transmitting, receiving and manipulating information on a computer system, the apparatus including a plurality of containers, each container being a logically defined data enclosure and comprising an information element, a plurality of registers, and a gateway. The plurality of registers, form part of the container, and include a first register for storing a unique container identification value; a second register having a representation designating time and governing interactions of the container with other containers, systems or processes according to utility of information in the information element relative to an external-to-the-apparatus event time; an active time register for identifying times at which the container will act upon other containers, processes, systems or gateways; a passive time register for identifying times at which the container can be acted upon by other containers, processes, systems or gateways; and a neutral time register for identifying times at which the container may interact with other containers, processes, systems or gateways. Additional registers designate space for container interactions.

PATENT APPLICATION FEE DETERMINATION RECORD
Effective Nov mber 10, 1998

Application or Docket Number
09/284113

CLAIMS AS FILED - PART I

SMALL ENTITY TYPE ☑ OR OTHER THAN SMALL ENTITY

	(Column 1)	(Column 2)	SMALL ENTITY			OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE	FEE		RATE	FEE
BASIC FEE				380.00	OR		~~760.00~~
TOTAL CLAIMS	36 minus 20=	* 16	X$ 9=	$144	OR	X$18=	
INDEPENDENT CLAIMS	3 minus 3 =	*	X39=		OR	X78=	
MULTIPLE DEPENDENT CLAIM PRESENT			+130=		OR	+260=	
			TOTAL	$524	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

8/10/04 **CLAIMS AS AMENDED - PART II**

AMENDMENT A	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY			OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 16	Minus	** 36	= —	X$ 9=		OR	X$18=	
Independent	* 2	Minus	*** 3	= —	X39=		OR	X78=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
					TOTAL ADDIT. FEE	Ø	OR	TOTAL ADDIT. FEE	

1/3/05

AMENDMENT B	(Column 1)		(Column 2)	(Column 3)					
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 16	Minus	** 36	=	X$ 9=		OR	X$18=	
Independent	* 4	Minus	*** 3	= 1	X39=		OR	X~~78~~88=	88
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	88

AMENDMENT C	(Column 1)		(Column 2)	(Column 3)					
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	*	Minus	**	=	X$ 9=		OR	X$18=	
Independent	*	Minus	***	=	X39=		OR	X78=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
☆ U.S. GPO:1998-454-473/90301

Scientific & Technical Information Center

USPTO

STIC Search Report

EIC 2100

STIC Database Tracking Number: 141562

TO: Cam-Linh T Nguyen Location: RND 3C21 Art Unit : 2161 Tuesday, January 04, 2005 Case Serial Number: 09/284113	From: David Holloway Location: EIC 2100 RND 4B19 Phone: 2-3528 david.holloway@uspto.gov

Search Notes

Dear Examiner Nguyen,

Attached please find your search results for above-referenced case.
Please contact me if you have any questions or would like a re-focused search.

David

sira
Search and Information Resources Administration

Access DB# ______

SEARCH REQUEST FORM

Scientific and Technical Information Center

Requester's Full Name: Nguyen, Cam Linh Examiner # : 78921 Date: 12/30/04
Art Unit: 2161 Phone Number 30 9-4024 Serial Number: 09/284,113
Mail Box and Bldg/Room Location: RND 3C21 Results Format Preferred (circle): [✓ PAPER] DISK E-MAIL

If more than one search is submitted, please prioritize searches in order of need.

****

Please provide a detailed statement of the search topic, and describe as specifically as possible the subject matter to be searched. Include the elected species or structures, keywords, synonyms, acronyms, and registry numbers, and combine with the concept or utility of the invention. Define any terms that may have a special meaning. Give examples or relevant citations, authors, etc, if known. Please attach a copy of the cover sheet, pertinent claims, and abstract.

Title of Invention: System & method for Creating and Manipulating Information containers with Dynamic Registers
Inventors (please provide full names): DeAngelo, Michael

Earliest Priority Filing Date: 1/30/98

***For Sequence Searches Only* Please include all pertinent information (parent, child, divisional, or issued patent numbers) along with the appropriate serial number.**

See claims

****

STAFF USE ONLY	Type of Search		Vendors and cost where applicable
Searcher: David Holloway	NA Sequence (#)		STN
Searcher Phone #: 2-3528	AA Sequence (#)		Dialog $ 721.4
Searcher Location: RND 4B19	Structure (#)		Questel/Orbit
Date Searcher Picked Up: 1-3-05	Bibliographic	✓	Dr.Link
Date Completed: 1-4-04	Litigation		Lexis/Nexis
Searcher Prep & Review Time: 50	Fulltext	✓	Sequence Systems
Clerical Prep Time:	Patent Family		WWW/Internet ✓
Online Time: 150	Other		Other (specify)

PTO-1590 (8-01)

```
Set      Items   Description
S1      272090   CONTAINER? OR ENVELOPE? OR BUCKET? OR (DATA OR INFORMATION-
              )()(ENCLOSURE? OR RECEPTACLE? OR FOLDER?)
S2      220678   REGISTER? OR REGISTR? OR (STORAGE OR MEMORY)(N)(LOCATION? -
              OR AREA OR AREAS OR ADDRESS? OR SECTOR? OR REGION?)
S3       19487   S2(2N)(MULTIPLE OR MULTIPLICITY OR PLURAL OR PLURALITY OR -
              MULTIPLICITY OR SEVERAL OR DIFFERENT OR MANY OR VARIOUS OR VA-
              RIETY)
S4         330   S3(5N)(ALTERABLE OR DYNAMIC? OR CHANGE? OR MODIFY? OR REVI-
              S? OR EDIT? OR LIVE OR HOT)
S5     1245365   TIME? OR SCHEDUL? OR HOUR? OR CALENDAR? OR TIMING OR TIMING
               OR DURATION? OR INTERVAL?
S6          87   S4(S)S5
S7          12   S1(10N)S3(10N)S5
S8           4   S1(S)S4(S)S5
S9          11   S6 AND IC=(G06F-017? OR G06F-007?)
S10         25   S7 OR S8 OR S9
S11         48   S6 AND IC=G06F?
S12         59   S10 OR S11
S13         42   S12 NOT AD=19980130:20010130
S14         36   S13 NOT AD=20010130:20030130
S15         35   S14 NOT AD=20030130:20050103
S16         11   S15 AND S1
S17         11   IDPAT (sorted in duplicate/non-duplicate order)
S18         11   IDPAT (primary/non-duplicate records only)
File 348:EUROPEAN PATENTS 1978-2004/Dec W03
         (c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20041230,UT=20041223
         (c) 2004 WIPO/Univentio
```

```
 18/3,K/2     (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00937647
System and method for parsing multiple sets of data
System und Verfahren zur Analyse mehrerer Datenmengen
Systeme et procede pour l'analyse de plusieurs ensembles de donnees
PATENT ASSIGNEE:
  Hewlett-Packard Company, (206030), 3000 Hanover Street, Palo Alto,
    California 94304, (US), (Applicant designated States: all)
INVENTOR:
  Pakenham, Gene, 5243 W 11th No. 1812, Greeley, Co 80634, (US)
  DeVore, Darwin A., 1221 Eastlake Court, Loveland, CO 80537, (US)
LEGAL REPRESENTATIVE:
  Carpmaels & Ransford (101821), 43 Bloomsbury Square, London WC1A 2RA,
    (GB)
PATENT (CC, No, Kind, Date):  EP 853418  A2  980715 (Basic)
                              EP 853418  A3  000705
APPLICATION (CC, No, Date):   EP 97309785 971204;
PRIORITY (CC, No, Date): US 782729 970113
DESIGNATED STATES: DE; FR; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: H04N-001/40; H04N-001/64
ABSTRACT WORD COUNT: 43
NOTE:
  Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:
Available Text  Language   Update    Word Count
      CLAIMS A  (English)  9829        351
      SPEC A    (English)  9829       5723
Total word count - document A         6074
Total word count - document B            0
Total word count - documents A + B    6074

...SPECIFICATION from the CCD cells into a sequential or serial data
  stream.
    A typical analog shift  register  comprises a  plurality  of "charge
  transfer  buckets " each of which is connected to an individual cell. At
  the end of the exposure  time , the charges collected by each of the CCD
  cells are simultaneously transferred to the charge transfer  buckets ,
  thus preparing the CCD cells for the next exposure sequence. The charge
  in each  bucket  is then transferred from  bucket  to  bucket  out of the
  shift register in a sequential or " bucket  brigade" fashion during the
  time the CCD cells are being exposed to the next scan...
 18/3,K/10     (Item 10 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00300850     **Image available**
UPDATE MECHANISM FOR COMPUTER STORAGE CONTAINER MANAGER
MOYEN  DE  MISE  A  JOUR  POUR  MODULE  DE  GESTION  D'ELEMENTS DE STOCKAGE
    D'ORDINATEURS
Patent Applicant/Assignee:
  APPLE COMPUTER INC,
Inventor(s):
  HARRIS Jared M,
  RUBEN Ira L,
Patent and Priority Information (Country, Number, Date):
  Patent:                 WO 9519001 A1 19950713
  Application:            WO 95US196 19950104  (PCT/WO US9500196)
  Priority Application: US 94177853 19940105
Designated States:
(Protection type is "patent" unless otherwise stated - for applications
prior to 2004)
  AM AT AU BB BG BR BY CA CH CN CZ DE DK EE ES FI GB GE HU JP KE KG KP KR
  KZ LK LR LT LU LV MD MG MN MW MX NL NO NZ PL PT RO RU SD SE SI SK TJ TT
  UA UZ VN KE MW SD SZ AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF
  BJ CF CG CI CM GA GN ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 119635

Fulltext Availability:
  Claims

Claim
...  the TOC and global name table of
  the target, Up to this point the updating  container
  has basically been opened "normally" like any other.
  It has its own TOC and global...

...that will eventually be returned to the user
  will be the one for the updating  container , so the
  targetFs TOC and global name table must be inherited
  by the updating CCB,
  The way the updating  container  gets to use the
   ...simply to copy
  the target's TOC and global name table pointers into
  the updating  container  CCB, But the pointers to the
  tables already there can't simply be clobbered. They...

...global name table pointers used by
  everyone, and the "private" pair mainly used by
  close- time  processing.
  One other pointer is inherited. That is a pointer
  referred to as the "target  container  pointer"
  (targetContainer), It is a CCB pointer copied from
  the target. It is always initialized...

...both "All and "B"s
  targetContainer will point to "B", This pointer is
  used for  container  refNum validity checks in the
  various API routines, It is the opposite of the
  updatingContainer pointer mentioned in step (1).
  updatingContainer points to the top-most  container ,
  and targetContainer the bottom-most (final or ultimate
  target).
  (5). Load in updater's non-private TOC
  If this is a previously existing updating
   container  opened for reading, then it is at this point
  all the updates from the updating  container  are
  applied to the target, The non-private portion of the
  updating  container 's TOC was loaded first in step (3),
```

Since the normal TOC is now the...
...updating" list properties for the objects they update will be encountered. As discussed for close- **time** processing, these will be value operations (set-infos, data edits, moves, etc.). The value data...

...represent all objects needing updating.
The touched chain can now be walked much like close- **time** processing to process the updating instructions associated with the "updating" property of each object on the chain. Also like close- **time** processing, objects on the touched chain are removed from the chain after each updating list...

...size and offsets are generated for the value headers and segments. Applying updates at this **time** changes the logical sizes and offsets. Thus, after each value's updates are completed, if...

...the value must be "re-logicalized".
It should also be pointed out that during this **time** , recording of updates is suppressed. It stays suppressed until the end of all open processing...

...and property updating instructions can be processed using the special TOC #1 property of the **container** 's private TOC.
At this point the target And the updating **container** have been opened, The updater's CCB pointer is returned to the user as the **container** refNum, The diagram of Fig. 22 illustrates the pertinent data structures discussed above. In the...

...target point to the same tables.
Since "All is opened first, then "B". the close- **time** processing reverses this by closing "B" then "A", In order to prevent the closing of...

...the
TOC and global name tables. This prevents premature release of the data.
H. Open- **time** Processing for Multi-layered Updaters
The previous discussion was mainly limited to one **container** updating another, It is fairly simple extension to the algorithm to allow for multiple updaters. Multiple updaters arise if a new **container** is opened for updating a target in multiple sessions. For example, the above situation was other hand, there is nothing preventing another new updating **container** from being opened,, say "XI', and using "All as its target; IIX updating "All updating...

...The process is the same, except for one variation in step (3) of the open- **time** processing; opening of the target **container** . Basically, as part of standard open- **time** processing, a check is always made to see if TOC #1 has a "Pointing value". This only exists in updating **container** TOCs and allows access to the proper target, be it separate or appended.
If the...

...TOC is present in memory that is indistinguishable from that of an ordinary, non-update **container** , except that some of the values refer to data actually present in other **containers** .
Each of the **containers** in the update chain remain open

so that value operations can reach the data,
I...

...Appendix D is a C-language header file for routines which read and write the **container** TOC. Appendix E sets forth the routines themselves, Appendix F is a C-language header file for a set of basic **container** handlers, and their metahandler, used by the **Container** Manager when doing update operations on a target **container** . Appendix G sets forth the handlers themselves.
The foregoing description of preferred embodiments of the...

...to
practitioners skilled in this art. As one example, an embodiment may permit an update **container** to be an update of two or more target **containers** concurrently, As another example, an implementation of the routines may construct only those aspects of the TOC in-memory which are needed for a particular operation after an update **container** is opened. The embodiments described herein were chosen and described in order to best explain...

```
Set     Items   Description
S1     905852   CONTAINER? OR ENVELOPE? OR BUCKET? OR (DATA OR INFORMATION-
             )()(ENCLOSURE? OR RECEPTACLE? OR FOLDER?)
S2    3635562   REGISTER? OR REGISTR? OR (STORAGE OR MEMORY)(N)(LOCATION? -
             OR AREA OR AREAS OR ADDRESS? OR SECTOR? OR REGION?)
S3      38232   S2(3N)(MULTIPLE OR MULTIPLICITY OR PLURAL OR PLURALITY OR -
             MULTIPLICITY OR SEVERAL OR DIFFERENT OR MANY OR VARIOUS OR VA-
             RIETY)
S4     142842   S2(5N)(ALTERABLE OR DYNAMIC? OR CHANGE? OR MODIFY? OR REVI-
             S? OR EDIT? OR LIVE OR HOT)
S5      50584   S1(12N)(TIME? OR SCHEDUL? OR HOUR? OR CALENDAR? OR TIMING -
             OR TIMING OR DURATION? OR INTERVAL?)
S6          0   S3(S)S4(S)S5
S7          6   S4(S)S5
S8          4   S3(S)S5
S9        570   S2(S)S5
S10       210   S2(10N)S5
S11         5   S10(10N)(ALTERABLE OR DYNAMIC? OR CHANGE? OR MODIFY? OR RE-
             VIS? OR EDIT? OR LIVE OR HOT)
S12        15   S11 OR S8 OR S7
S13         9   RD (unique items)
S14       146   S2(5N)S5
S15        57   S14(S)(DATA OR BIT OR BYTE? OR DATABLOCK? OR INFORMATION? -
             OR MEMOR? OR STORAGE? OR BUFFER? OR CACHE?)
S16        37   RD (unique items)
S17        45   S16 OR S13
S18        45   RD (unique items)
S19        23   S18 NOT PY>1998
S20        21   S19 NOT PD=19980130:20010130
S21        21   S20 NOT PD=20010130:20050110
File 275:Gale Group Computer DB(TM) 1983-2005/Jan 04
         (c) 2005 The Gale Group
File  47:Gale Group Magazine DB(TM) 1959-2005/Jan 04
         (c) 2005 The Gale group
File  75:TGG Management Contents(R) 86-2004/Dec W1
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2005/Jan 04
         (c) 2005 The Gale Group
File  16:Gale Group PROMT(R) 1990-2005/Jan 04
         (c) 2005 The Gale Group
File 624:McGraw-Hill Publications 1985-2004/Dec 28
         (c) 2004 McGraw-Hill Co. Inc
File 484:Periodical Abs Plustext 1986-2004/Dec W4
         (c) 2004 ProQuest
File 613:PR Newswire 1999-2005/Jan 03
         (c) 2005 PR Newswire Association Inc
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 141:Readers Guide 1983-2004/Sep
         (c) 2004 The HW Wilson Co
File 370:Science 1996-1999/Jul W3
         (c) 1999 AAAS
File 696:DIALOG Telecom. Newsletters 1995-2005/Jan 03
         (c) 2005 The Dialog Corp.
File 553:Wilson Bus. Abs. FullText 1982-2004/Sep
         (c) 2004 The HW Wilson Co
File 621:Gale Group New Prod.Annou.(R) 1985-2005/Jan 04
         (c) 2005 The Gale Group
File 674:Computer News Fulltext 1989-2004/Dec W2
         (c) 2004 IDG Communications
File  88:Gale Group Business A.R.T.S. 1976-2005/Dec 30
         (c) 2005 The Gale Group
File 369:New Scientist 1994-2004/Dec W3
         (c) 2004 Reed Business Information Ltd.
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 635:Business Dateline(R) 1985-2005/Jan 01
         (c) 2005 ProQuest Info&Learning
```

```
File  15:ABI/Inform(R) 1971-2005/Jan 01
         (c) 2005 ProQuest Info&Learning
File   9:Business & Industry(R) Jul/1994-2005/Jan 03
         (c) 2005  The Gale Group
File  13:BAMP 2005/Dec W4
         (c) 2005  The Gale Group
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 610:Business Wire 1999-2005/Jan 03
         (c) 2005 Business Wire.
File 647:CMP  Computer Fulltext 1988-2005/Dec W3
         (c) 2005 CMP Media, LLC
File  98:General Sci Abs/Full-Text 1984-2004/Sep
         (c) 2004 The HW Wilson Co.
File 148:Gale Group Trade & Industry DB 1976-2004/Jan 03
         (c)2004 The Gale Group
File 634:San Jose Mercury  Jun 1985-2004/Dec 31
         (c) 2005 San Jose Mercury News
```

```
Set     Items   Description
S1     325405   CONTAINER? OR ENVELOPE? OR BUCKET? OR (DATA OR INFORMATION-
             )()(ENCLOSURE? OR RECEPTACLE? OR FOLDER?)
S2     304185   REGISTER? OR REGISTR? OR (STORAGE OR MEMORY)(N)(LOCATION? -
             OR AREA OR AREAS OR ADDRESS? OR SECTOR? OR REGION?)
S3       5930   S2(2N)(MULTIPLE OR MULTIPLICITY OR PLURAL OR PLURALITY OR -
             MULTIPLICITY OR SEVERAL OR DIFFERENT OR MANY OR VARIOUS OR VA-
             RIETY)
S4         91   S3(5N)(ALTERABLE OR DYNAMIC? OR CHANGE? OR MODIFY? OR REVI-
             S? OR EDIT? OR LIVE OR HOT)
S5    8070002   TIME? OR SCHEDUL? OR HOUR? OR CALENDAR? OR TIMING OR TIMING
              OR DURATION? OR INTERVAL?
S6          1   S1 AND S4
S7      56519   S1 AND S5
S8         11   S3 AND S7
S9         36   S1 AND S3
S10        36   S6 OR S8 OR S9
S11        27   RD (unique items)
S12       445   S1 AND S2 AND S5
S13      8654   S1(3N)(DATA OR INFORMATION OR BYTE? OR BITS OR MEGABYTE? OR
              KILOBYTE? OR STORAGE? OR MEMORY?)
S14        49   S12 AND S13
S15        74   S14 OR S11
S16        65   RD (unique items)
S17        49   S16 NOT PY>1998
File   8:Ei Compendex(R) 1970-2005/Dec W4
         (c) 2005 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2004/Dec
         (c) 2004 ProQuest Info&Learning
File  65:Inside Conferences 1993-2004/Dec W4
         (c) 2004 BLDSC all rts. reserv.
File   2:INSPEC 1969-2004/Dec W2
         (c) 2004 Institution of Electrical Engineers
File  94:JICST-EPlus 1985-2004/Nov W4
         (c)2004 Japan Science and Tech Corp(JST)
File 111:TGG Natl.Newspaper Index(SM) 1979-2004/Dec 29
         (c) 2004 The Gale Group
File   6:NTIS 1964-2004/Dec W4
         (c) 2004 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2004/Dec W1
         (c) 2004 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2004/Dec W4
         (c) 2004 Inst for Sci Info
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Nov
         (c) 2004 The HW Wilson Co.
File  95:TEME-Technology & Management 1989-2004/Jun W1
         (c) 2004 FIZ TECHNIK
```

17/5/13 (Item 13 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

00267759 E.I. Monthly No: EI7212010596
Title: SIMPLE CHARGE REGENERATOR FOR USE WITH CHARGE-TRANSFER DEVICES AND THE DESIGN OF FUNCTIONAL LOGIC ARRAYS.
Author: Tompsett, Michael F.
Corporate Source: Bell Telephone Lab, Inc, Murray Hill, NJ
Source: IEEE Journal of Solid-State Circuits v SC-7 n 3 Jun 1972 p 237-242
Publication Year: 1972
CODEN: IJSCBC ISSN: 0018-9200
Language: ENGLISH
Journal Announcement: 7212

Abstract: An inverting binary-charge regenerator for use with new charge-transfer devices (charge-coupled and integrated MOS **bucket** brigade) is described. This simple element requires an area approximately that of one bit in the register and is driven by the transfer pulses. Its uses with these shift **registers** in **various** configurations, which are described, make possible even larger functional devices. These uses include regeneration in serial memories, performing logic operations such as NAND and NOR involving the bit trains in **several** **registers** , and performing fixed counts and sequential addressing of other circuit elements.

Descriptors: *LOGIC CIRCUITS
Classification Codes:
721 (Computer Circuits & Logic Elements)
72 (COMPUTERS & DATA PROCESSING)

17/5/17   (Item 4 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01219256 ORDER NO: AAD92-14527
ANALYTICAL MODELS AND OPTIMAL STRATEGIES FOR AUTOMATED STORAGE/RETRIEVAL SYSTEM OPERATIONS (STORAGE-RETRIEVAL)

Author: PARK, BYUNG CHUN
Degree: PH.D.
Year: 1991
Corporate Source/Institution: GEORGIA INSTITUTE OF TECHNOLOGY (0078)
Director: EDWARD H. FRAZELLE
Source: VOLUME 52/12-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
         PAGE 6592. 184 PAGES
Descriptors: ENGINEERING, INDUSTRIAL
Descriptor Codes: 0546

The objective of this research is to provide exact reliable expressions for use in designing and operating automated storage/retrieval systems. We focus on the efficient operation of dual command cycles. The main performance criteria are travel **time** and system throughput.

We begin by developing a general analytical baseline for automated storage/retrieval system performance analysis. The baseline is a closed-form expression for the mean and variance of single and dual command cycle **time** . The model can be effectively used for evaluating any storage policy, i.e., random, dedicated or class-based storage. We give examples to illustrate how the formulas can be used for evaluating each storage policy.

With an evaluation baseline, our attention turns to specific improvement strategies. Contour line configurations for **storage location** assignment are developed first. We develop a general scheme to generate contour line configurations for dual command operations. To investigate the effects of alternative contour line configurations on system performance, a series of experiments are performed. The storage policies considered are random storage, priority-based open location (POL) storage, turnover-based storage, and 2-class storage. The performance of each contour line configuration is measured in terms of the expected dual command travel **time** .

Next, we develop optimal dwell point policies for automated storage/retrieval systems. Based on the fact that dwell point policies minimize the completion **time** of the first transaction after the storage/retrieval machine becomes idle, we show that there is a unique optimal dwell point policy, regardless of other system parameters. Then a variety of return paths to the dwell point are introduced and studied.

Finally, an end-of-aisle order picking system with inbound and outbound buffer positions is studied. This is usually referred to as a miniload system with a horse-shoe "front-end". The system is modeled as a two-stage cyclic queueing system consisting of one general and one exponential server with limited capacity. The cyclic queueing system is then analyzed by using the customer-hole duality concept. Closed-form expressions for the stationary probability and system throughput are developed. We also obtain the proportion of picker-idle and storage/retrieval machine-idle **time** by noting that the throughput of the picker is equal to that of the storage/retrieval machine. Then, a design problem to determine the optimal number of inbound and outbound buffer positions and a control problem to determine the optimal number of **storage containers** in the system are studied. The effect of buffer size on system throughput is also investigated.

```
Set     Items   Description
S1     675611   CONTAINER? OR ENVELOPE? OR BUCKET? OR (DATA OR INFORMATION-
             )()(ENCLOSURE? OR RECEPTACLE? OR FOLDER?)
S2     397825   REGISTER? OR REGISTR? OR (STORAGE OR MEMORY)(N)(LOCATION? -
             OR AREA OR AREAS OR ADDRESS? OR SECTOR? OR REGION?)
S3      12040   S2(2N)(MULTIPLE OR MULTIPLICITY OR PLURAL OR PLURALITY OR -
             MULTIPLICITY OR SEVERAL OR DIFFERENT OR MANY OR VARIOUS OR VA-
             RIETY)
S4        140   S3(5N)(ALTERABLE OR DYNAMIC? OR CHANGE? OR MODIFY? OR REVI-
             S? OR EDIT? OR LIVE OR HOT)
S5    3326894   TIME? OR SCHEDUL? OR HOUR? OR CALENDAR? OR TIMING OR TIME(-
             )STAMP?
S6         34   S4 AND S5
S7          0   S1 AND S4
S8         70   S1 AND S3
S9          5   S8 AND IC=(G06F-017? OR G06F-007?)
S10        11   S8 AND IC=G06F?
S11        20   S4 AND IC=(G06F-017? OR G06F-007?)
S12        22   S6 AND IC=G06F?
S13     88865   MC=(T01-C04? OR T01-J05B?)
S14         5   S13 AND (S6 OR S8)
S15        40   S12 OR S11 OR S14
S16        40   IDPAT (sorted in duplicate/non-duplicate order)
S17        39   IDPAT (primary/non-duplicate records only)
File 347:JAPIO Nov 1976-2004/Aug(Updated 041203)
         (c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200482
         (c) 2004  Thomson Derwent
```

```
 17/5/3     (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004  Thomson Derwent. All rts. reserv.

015947975     **Image available**
WPI Acc No: 2004-105816/200411
  Method and system for vector  scheduling  on object code level
Patent Assignee: UNIV INHA (UYIN-N)
Inventor: KIM G C; KIM J S; KIM S D; LEE D H; LEE Y S
Number of Countries: 001  Number of Patents: 001
Patent Family:
Patent No      Kind   Date     Applicat No     Kind   Date      Week
KR 2003078467  A   20031008  KR 200217526    A   20020329  200411  B

Priority Applications (No Type Date): KR 200217526 A 20020329
Patent Details:
Patent No  Kind Lan Pg   Main IPC    Filing Notes
KR 2003078467 A      1 G06F-009/38

Abstract (Basic): KR 2003078467 A
        NOVELTY - A method and a system for vector  scheduling  on an
    object code level are provided to produce an excellent  scheduling
    result in a  scheduling   time  faster than a present software
    pipelining method.
        DETAILED DESCRIPTION - An LCV(Loop Control Variable), and an
    initialization instruction, an initial value, a change instruction, a
    variance width, and an end instruction of the LCV are found out by
    using a CFG(Control Flow Graph) of a vector loop as input. The number
    of the concurrent executing instructions is calculated and the copies
    are generated by developing the vector loop with x. Registers are
    renamed in order to remove the data dependency of the registers used
    for each copy. The LCV of the copies is  changed  to the  register
    different  with each other. The LCV initialization instruction in the
    copies is changed based on the variance width. The LCV changing
    instruction in the copies is changed based on the x value. The
    instruction for copying the value of the LCV to the variable is
    inserted into a loop termination path of the copies. The vector
    scheduled  CFG is generated by collecting the duplicated instruction
    for each instruction of the first copy.
        pp; 1 DwgNo 1/10
Title Terms: METHOD; SYSTEM; VECTOR;  SCHEDULE ; OBJECT; CODE; LEVEL
Derwent Class: T01
International Patent Class (Main):  G06F-009/38
File Segment: EPI
```

```
 17/5/5     (Item 5 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004  Thomson Derwent. All rts. reserv.

015768651     **Image available**
WPI Acc No: 2003-830853/200377
XRPX Acc No: N03-663897
  Dynamic random access memory controller in computer system, has
  configuration registers to store control information of memory banks, and
  column address strobe state machine to generate strobe signals for memory
  banks
Patent Assignee: INTEL CORP (ITLC  )
Inventor: LANGENDORF B K; DODD J M; WADE N D
Number of Countries: 001 Number of Patents: 002
Patent Family:
Patent No      Kind   Date     Applicat No     Kind   Date      Week
US 20030177303  A1  20030918  US 97814697      A   19970311  200377  B
                              US 2003389092   A   20030313
US 6725349     B2  20040420  US 94381091      A   19941223  200427
                              US 97814697     A   19970311
                              US 2003389092   A   20030313

Priority Applications (No Type Date): US 97814697 A 19970311; US 2003389092
  A 20030313; US 94381091 A 19941223
Patent Details:
Patent No  Kind Lan Pg   Main IPC    Filing Notes
US 20030177303 A1    10 G06F-012/00   Cont of application US 97814697
US 6725349    B2       G06F-012/00   Cont of application US 94381091
                                     Cont of application US 97814697

Abstract (Basic): US 20030177303 A1
        NOVELTY -  Several  configuration  registers  (300) store control
    information for  dynamic  RAM (DRAM) memory banks of a main memory
    (103). A column address strobe (CAS) state machine (330) coupled to the
    registers, generates CAS signals (220) for the memory banks. A
    detection logic circuit coupled to the CAS state machine, determines
    type of DRAM device installed in each memory bank to store control
    information of the device in the registers.
        DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for
    method for optimizing control of each memory bank.
        USE - For automatically configuring and controlling memory banks
    installed with dynamic random access memory (DRAM) devices including
    standard page mode DRAM and extended data-out DRAM (EDO-DRAM) in
    computer system.
        ADVANTAGE - The CAS state machine automatically controls  timing
    requirements of the DRAM devices installed in the main memory to
    quickly and efficiently handle access requests. Thus, the performance
    of EDO-DRAM and standard page mode DRAM is preserved and controlled
    without increasing the hardware cost, while ensuring correct operation
    of the DRAMs.
        DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of
    the DRAM controller.
        main memory (103)
        CAS signals (220)
        configuration register (300)
        address bank decoder (310)
        CAS state machine (320)
        pp; 10 DwgNo 3/7
Title Terms: DYNAMIC; RANDOM; ACCESS; MEMORY; CONTROL; COMPUTER; SYSTEM;
  CONFIGURATION; REGISTER; STORAGE; CONTROL; INFORMATION; MEMORY; BANK;
  COLUMN; ADDRESS; STROBE; STATE; MACHINE; GENERATE; STROBE; SIGNAL; MEMORY
  ; BANK
Derwent Class: T01; U14
International Patent Class (Main):  G06F-012/00
File Segment: EPI
```

17/5/11 (Item 11 from file: 350)
DIALOG(R)File 350:Derwent WPIX

013483279 **Image available**
WPI Acc No: 2000-655222/200063
Related WPI Acc No: 1999-457716
XRPX Acc No: N00-485639

Data processing system used in graphical user interface, displays one of subset of stored object automatically only if at least one of stored object of subset has not been manually associated with its container

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: LISLE L A; MARTIN S L; MULLALY J M
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6104394	A	20000815	US 97813717	A	19970307	200063	B
			US 99239405	A	19990128		

Priority Applications (No Type Date): US 97813717 A 19970307; US 99239405 A 19990128
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6104394	A		19	G06F-017/30	Cont of application US 97813717 Cont of patent US 5936624

Abstract (Basic): US 6104394 A

NOVELTY - Two subsets constituting stored object of respective object types within respective **containers** , are displayed automatically in a display unit, only if at least one of stored objects of the two subsets has not been manually associated with their respective **containers** .

DETAILED DESCRIPTION - The display unit automatically updates the displayed subsets when the stored objects is charged, when either storing, editing, deleting, moving, archiving, copying, linking on undoing of stored object occurs. The object type is chosen from text type, audio type, graphic type, type corresponding to date and time.
INDEPENDENT CLAIMS are also included for the following:

(a) operating data processing system;

(b) program product

USE - Used in graphical user interface, real world style interface with logical containment system.

ADVANTAGE - Enables user to organize representation of desired objects in **various storage locations** without requiring extra steps by a user access memory. Allows greater flexibility in obtaining desired graphical user interface, by the ability of user to modify the containment settings and to have modifications immediately rejected in logical **container** rendered on display device.

DESCRIPTION OF DRAWING(S) - The figure shows model diagram of data processing system explained with Booch notation.

pp; 19 DwgNo 8/8

Title Terms: DATA; PROCESS; SYSTEM; GRAPHICAL; USER; INTERFACE; DISPLAY; ONE; SUBSET; STORAGE; OBJECT; AUTOMATIC; ONE; STORAGE; OBJECT; SUBSET; MANUAL; ASSOCIATE; **CONTAINER**
Derwent Class: T01
International Patent Class (Main): G06F-017/30
File Segment: EPI

```
 17/5/22     (Item 22 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004  Thomson Derwent. All rts. reserv.

007659045     **Image available**
WPI Acc No: 1988-292977/198841
XRPX Acc No: N88-222362
  Multiprocessor system with shared memory - has machine instruction
  sequence in shared memory for assigning register sets based on status
  information
Patent Assignee: STELLAR COMPUTER IN (STEL-N)
Inventor: DARNELL P A; MORTON M A
Number of Countries: 029  Number of Patents: 002
Patent Family:
Patent No     Kind   Date    Applicat No    Kind   Date     Week
WO 8807720     A   19881006  WO 88US1032     A   19880325  198841  B
AU 8816821     A   19881102                                198904

Priority Applications (No Type Date): US 8734166 A 19870402
Cited Patents: 1.Jnl.Ref; EP 174446; US 3916383; US 3972029; US 3980922; US
  4121286; US 4197579; US 4280176; US 4354227; US 4713757
Patent Details:
Patent No  Kind Lan Pg   Main IPC    Filing Notes
WO 8807720    A  E  27
   Designated States (National): AU BB BG BR DK FI HU JP KP KR LK MC MG MW
   NO RO SD SU
   Designated States (Regional): AT BE CH DE FR GB IT LU NL OA SE

Abstract (Basic): WO 8807720 A
        The multiprocessor system has four processors which share machine
    instruction sequence (14) stored in parallel regions in the memory
    (16), which includes parallel regions of instructions. Each region has
    two blocks of instructions which are independent in that the same
    result is obtained if the blocks are executed by the same processor or
    by different processors.
        The system provides a pool (22) of high speed register sets to
    regulate the work of the four processors. Each processor has a
    respective unshared register (40) which stores a program status word.
    The word provides information about the part of the process executing
    on a given processor at a given  time .
        ADVANTAGE - The assignment of shared  register  sets to  multiple
    processors is effected  dynamically  without interrupting the execution
    of instructions in the sequence.
        1/8
Title Terms: MULTIPROCESSOR; SYSTEM; SHARE; MEMORY; MACHINE; INSTRUCTION;
  SEQUENCE; SHARE; MEMORY; ASSIGN; REGISTER; SET; BASED; STATUS;
  INFORMATION
Derwent Class: T01
International Patent Class (Additional):  G06F-012/00
File Segment: EPI
```

17/5/25 (Item 25 from file: 347)
DIALOG(R)File 347:JAPIO

07054897 **Image available**
COMMUNICATION INSTRUCTION RESULT OF PROCESSOR AND COMPILING METHOD FOR PROCESSOR

PUB. NO.:	2001-282532 [JP 2001282532 A]
PUBLISHED:	October 12, 2001 (20011012)
INVENTOR(s):	TOPHAM NIGEL PETER
APPLICANT(s):	SIROYAN LTD
APPL. NO.:	2001-032090 [JP 200132090]
FILED:	February 08, 2001 (20010208)
PRIORITY:	00 200002848 [GB 20002848], GB (United Kingdom), February 08, 2000 (20000208)
INTL CLASS:	**G06F-009/38 ; G06F-009/30 ; G06F-009/34 ; G06F-009/45**

ABSTRACT

PROBLEM TO BE SOLVED: To simplify a task of a compiler for the allocation of a register and to set up an instruction in a more compact state.

SOLUTION: A processor 1 for executing a pipeline by software includes an instruction issuing device 10 for issuing plural instructions to be executed by a previously determined sequence. The sequence of instructions includes plural value generation instructions for generating respective values at the **time** of execution of the sequence. Each of instruction execution devices 14, 16, 18 executes an issued instruction. A register file 20 has plural registers and stores plural values generated by respective executed instructions. During the period of operation, the processor 1 allocates plural values generated by respective value generation instructions to respective sequence numbers in accordance with the issued order of respective value generation instructions. Each generated value is allocated to one of plural registers in order to store the generated value on the basis of the sequence number allocated to the value. The names of these **plural registers** can be **changed** in each issue of a value generation instruction.

17/5/31 (Item 31 from file: 347)
DIALOG(R)File 347:JAPIO

03475828 **Image available**
DIGITAL PROCESSOR

PUB. NO.:	03-138728 [JP 3138728 A]
PUBLISHED:	June 13, 1991 (19910613)
INVENTOR(s):	SATOMURA RYUICHI TOMOBE KATSUICHI
APPLICANT(s):	HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.:	01-275821 [JP 89275821]
FILED:	October 25, 1989 (19891025)
INTL CLASS:	[5] **G06F-009/38**
JAPIO CLASS:	45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JAPIO KEYWORD:	R131 (INFORMATION PROCESSING -- Microcomputers & Microprocessers)
JOURNAL:	Section: P, Section No. 1250, Vol. 15, No. 360, Pg. 105, September 11, 1991 (19910911)

ABSTRACT

PURPOSE: To shorten the instruction execution **time** of a microprocessor, etc., and to raise processing capacity by **dynamically** assigning **plural** work **registers** in accordance with its usage condition.

CONSTITUTION: n numbers of work registers WR1-WRn used when execution a micro-instruction, and a work register control part WRC which dynamically assigns these work registers WR1-WRn in accordance with their usage condition, are provided. Then, the work registers WR1-WRn can be dynamically assigned in accordance with their usage condition without specifying them with a micro-instruction, the score boarding of the work registers WR1-WRn can be executed simultaneously with the score boarding of a general purpose register GR by a machine language instruction, and the parallel processing of he following machine language instructions can be started early. Thus, the instruction execution **time** of the microprocessor, etc., can be shortened equally, and its processing capacity can be raised.

17/5/36 (Item 36 from file: 347)
DIALOG(R)File 347:JAPIO

01917037 **Image available**
CONTINUOUS GENERATING SYSTEM OF PLURAL ADDRESSES

PUB. NO.:	61-131137 [JP 61131137 A]
PUBLISHED:	June 18, 1986 (19860618)
INVENTOR(s):	AKIBA HIROSHI AOYANAGI KEIZO
APPLICANT(s):	TOSHIBA CORP [000307] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.:	59-253342 [JP 84253342]
FILED:	November 30, 1984 (19841130)
INTL CLASS:	[4] **G06F-012/02**
JAPIO CLASS:	45.2 (INFORMATION PROCESSING -- Memory Units)
JOURNAL:	Section: P, Section No. 512, Vol. 10, No. 324, Pg. 91, November 05, 1986 (19861105)

ABSTRACT

PURPOSE: To attain count-up or count-down sequentially for **plural** address **registers** by using an adder to **modify** logically an address of an address register before one address and applying it sequentially to each register.

CONSTITUTION: An address modification data and a logical address from an adder 2 inputting an addend for modification and generating a logical address are inputted sequentially to plural address registers 3,4,5. Then address information of the address registers 3,4,5 is selected in the predetermined order by a selector 10 and outputs it as a memory access (f). Further, the address register selected precedingly by the selector 10 is selected by the other selector 9 at the same **time** and the address information (e) is fed back to a selector 6 as the address modification data. Then a required addition is executed by the adder 2 to modify the address of the register subject to feed back. The operation is executed sequentially to the registers 3,4,5 to generate continuously plural addresses thereby clearing them.

17/5/37 (Item 37 from file: 347)
DIALOG(R)File 347:JAPIO

01655844 **Image available**
GUIDANCE INFORMATION CONTROLLING SYSTEM

PUB. NO.: 60-134344 [JP 60134344 A]
PUBLISHED: July 17, 1985 (19850717)
INVENTOR(s): YOSHINO ISAO
SOMA MASATO
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 58-241921 [JP 83241921]
FILED: December 23, 1983 (19831223)
INTL CLASS: [4] **G06F-009/00** ; **G06F-015/00**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units); 45.4 (INFORMATION PROCESSING -- Computer Applications)
JOURNAL: Section: P, Section No. 408, Vol. 09, No. 299, Pg. 50, November 27, 1985 (19851127)

ABSTRACT

PURPOSE: To execute a change which requires no person's help by executing the change by sending guidance information to a terminal control device from a central processor by a guidance information change request from a change request mechanism of the central processor or a terminal equipment.

CONSTITUTION: A central processor 11, terminal control device 16 and a terminal equipment 20 are connected, a local guidance control mechanism 12 and a **timer** 13 are provided on the processor 11, and a storage mechanism 15 and a managing mechanism 14 are provided on the mechanism 12. Also, a local guidance change control mechanism 19 consisting of a store part 17 and a change identifying mechanism 18 is provided on the device 16, and the store part 17 is constituted of **plural storage area** 21 and on-demand **changeable** storage areas 22. In this state, the area 21 is changed by receiving a change request from the **timer** 13 by the mechanism 14 and sending retrieved information to the mechanism 19, a change request from the equipment 20 is registered in the mechanism 18 and also transferred to the processor 11, and the area 22 is changed by sending the retrieved information to the mechanism 19.

17/5/38 (Item 38 from file: 347)
DIALOG(R)File 347:JAPIO

01125342 **Image available**
OPERATION PROCESSING DEVICE

PUB. NO.:	58-062742 [JP 58062742 A]
PUBLISHED:	April 14, 1983 (19830414)
INVENTOR(s):	TAMURA NOBORU
APPLICANT(s):	CANON INC [000100] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.:	56-162367 [JP 81162367]
FILED:	October 12, 1981 (19811012)
INTL CLASS:	[3] **G06F-007/00**
JAPIO CLASS:	45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL:	Section: P, Section No. 208, Vol. 07, No. 152, Pg. 39, July 05, 1983 (19830705)

ABSTRACT

PURPOSE: To make the operation easy, by providing a means having a **plurality** of condition **registers** and storing the **change** in the condition codes in the past, and a means performing the operation instruction between condition registers.
CONSTITUTION: In executing an arithmetic logical operation instruction having the alteration of a CCR through the instruction of an instruction decoder 6, the conditions with the result of operation are set to the CCR. An arithmetic logical operation device 3 reads out the 1st CCR4-1 in one machine cycle and gives an output to the 2nd CCR4-2 in the next machine cycle. The conditions through the result of present operation are outputted to the 1st CCR4-1 in the next machine cycle. Through this operation, the previous condition is set to the 2nd CCTR4-2 and the present condition is set to the 1st CCR4-1. Further, with the operation instruction between the CCRs, the two CCRs 4-1, 4-2 are read and an output is given to the instructed CCR

```
Set      Items   Description
S1           9   AU=(DEANGELO, M? OR DEANGELO M? OR DE ANGELO M? OR DE ANGE-
              LO, M?)
S2           3   S1 AND IC=G06F?
File 347:JAPIO Nov 1976-2004/Aug(Updated 041203)
         (c) 2004 JPO & JAPIO
File 348:EUROPEAN PATENTS 1978-2004/Dec W03
         (c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20041230,UT=20041223
         (c) 2004 WIPO/Univentio
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200482
         (c) 2004  Thomson Derwent
```

```
2/5/1     (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

01077982
SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH
    DYNAMIC REGISTERS
SYSTEM   UND   VERFAHREN   ZUR   ERZEUGUNG   UND   BEARBEITUNG   VON
    INFORMATIONSBEHALTERN MITDYNAMISCHEN REGISTERN.
SYSTEME  ET  PROCEDE  POUR  LA  CREATION  ET  LA MANIPULATION DE CONTENEURS
    D'INFORMATIONS A REGISTRES DYNAMIQUES
PATENT ASSIGNEE:
  Ematrix Corporation, (2819080), 104 West Anapamu, Santa Barbara, CA 93101
    , (US), (Applicant designated States: all)
INVENTOR:
   De Angelo, Michael , Suite 290, 1324 J State Street, Santa Barbara, CA
    93101, (US
LEGAL REPRESENTATIVE:
  McLeish, Nicholas Alistair Maxwell et al (74621), Boult Wade Tennant
    Verulam Gardens 70 Gray's Inn Road, London WC1X 8BT, (GB)
PATENT (CC, No, Kind, Date):  EP 1049996  A1  001108 (Basic)
                              WO 9939285  990805
APPLICATION (CC, No, Date):   EP 99905548 990128;  WO 99US1988  990128
PRIORITY (CC, No, Date): US 73209 980130
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI;
  LU; MC; NL; PT; SE
INTERNATIONAL PATENT CLASS:  G06F-017/30 ;  G06F-003/14
NOTE:
  No A-document published by EPO
LEGAL STATUS (Type, Pub Date, Kind, Text):
 Application:      001108 A1 Published application with search report
 Application:      991006 A1 International application. (Art. 158(1))
 Withdrawal:       040204 A1 Date application deemed withdrawn: 20030801
 Examination:      001108 A1 Date of request for examination: 20000713
 Search Report:    010425 A1 Date of drawing up and dispatch of
                             supplementary:search report 20010308
 Application:      991006 A1 International application entering European
                             phase
LANGUAGE (Publication,Procedural,Application): English; English; English


 2/5/2     (Item 1 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00507933    **Image available**
SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH
    DYNAMIC REGISTERS
SYSTEME  ET  PROCEDE  POUR  LA  CREATION  ET  LA MANIPULATION DE CONTENEURS
    D'INFORMATIONS A REGISTRES DYNAMIQUES
Patent Applicant/Assignee:
  EMATRIX CORPORATION,
  DE ANGELO Michael,
Inventor(s):
   DE ANGELO Michael
Patent and Priority Information (Country, Number, Date):
  Patent:                WO 9939285 A1 19990805
  Application:           WO 99US1988 19990128  (PCT/WO US9901988)
  Priority Application: US 9873209 19980130
Designated States:
```

(Protection type is "patent" unless otherwise stated - for applications prior to 2004)

AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GD GE GH
GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN
MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN YU
ZW GH GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE
DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR
NE SN TD TG

Main International Patent Class: **G06F-017/30**
International Patent Class: **G06F-003/14**
Publication Language: English
Fulltext Availability:
  Detailed Description
  Claims
Fulltext Word Count: 18390

English Abstract

A system for creating and manipulating information containers with dynamic registers on a multi-user computer system, or computer network comprises an interactive information container, a container editor, a search interface, a user profile, system-wide hierarchical container gateways (site 7), interactive and evolving container registers, a data collection means, a data reporting means, an analysis engine with editor, an executing engine with editor, and a means of communicating with other computers, computer networks, or digital-based public or published media. The container editor provides an authoring user with the capacity to encapsulate any information component such as a file, set, database, network, event or process, and a set of parameters of multiple container registers to govern the interaction of that container with other containers or processes. The container registers include system-defined, system-alterable, user-defined and user-alterable registers.

French Abstract

L'invention concerne un systeme pour la creation et la manipulation de conteneurs d'informations a registres dynamiques, sur un systeme informatique multi-utilisateur, ou sur un reseau informatique. Ce systeme comprend un conteneur d'informations interactif, un editeur de conteneur, une interface de recherche, un profil d'utilisateur, des passerelles (site 7) de conteneurs hierarchiques a l'echelle du systeme, des registres interactifs et evolutifs, un dispositif de rassemblement de donnees, un dispositif d'edition de donnees, un moteur d'analyse avec editeur, un moteur d'execution avec editeur, et un dispositif permettant de communiquer avec d'autres ordinateurs, avec des reseaux informatiques, ou avec des supports numeriques publics ou publies. L'editeur de conteneur permet a un utilisateur-auteur d'encapsuler n'importe quel composant d'information tel qu'un dossier, un ensemble, une base de donnees, un reseau, un evenement ou un procede, et fournit a cet utilisateur une serie de parametres pour plusieurs registres de conteneurs pour commander l'interaction de ce conteneur avec d'autres conteneurs ou procedes. Les registres de conteneurs comprennent des registres definis par le systeme, modifiables par le systeme, definis par l'utilisateur et modifiables par l'utilisateur.

2/5/3 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012673114 **Image available**
WPI Acc No: 1999-479221/199940
XRPX Acc No: N99-356757

Computer system for creating and manipulating information containers in multi-user systems e.g. client server network

Patent Assignee: EMATRIX CORP (EMAT-N)

Inventor: **DE ANGELO M**

Number of Countries: 085 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9939285	A1	19990805	WO 99US1988	A	19990128	199940	B
AU 9925687	A	19990816	AU 9925687	A	19990128	200002	
EP 1049996	A1	20001108	EP 99905548	A	19990128	200062	
			WO 99US1988	A	19990128		

Priority Applications (No Type Date): US 9873209 P 19980130

Patent Details:

```
Patent No  Kind Lan Pg   Main IPC     Filing Notes
WO 9939285    A1 E  78 G06F-017/30
   Designated States (National): AL AM AT AU AZ BA BB BG BR BY CA CH CN CU
   CZ DE DK EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC
   LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL
   TJ TM TR TT UA UG US UZ VN YU ZW
   Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR
   IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW
AU 9925687    A          G06F-017/30   Based on patent WO 9939285
EP 1049996    A1 E       G06F-017/30   Based on patent WO 9939285
   Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI
   LU MC NL PT SE
```

Abstract (Basic): WO 9939285 A1

NOVELTY - A container editor provides an authoring user with the capacity to encapsulate any information component such as a file, set, database, network, event or process, and a set of parameters of multiple container registers to govern the interaction of that container with other containers or processes. The container-registers include system-defined, system-alterable, user-definable and user-alterable algorithms.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for; a method for creating an interactive information container; a method for interacting between two interactive information containers.

USE - Transmitting, receiving and manipulating information containers with dynamic registers on a multi-user computer system, or computer network e.g. local, wide area or public networks, in computer, media or publishing networks.

ADVANTAGE - Information can be manufactured on, utility upgraded, and intelligence developed in a computer network by offering the mechanism to create and manipulate information containers with dynamic registers.

DESCRIPTION OF DRAWING(S) - The drawing shows a block diagram of a preferred embodiment of the invention.

pp; 78 DwgNo 1/17

Title Terms: COMPUTER; SYSTEM; MANIPULATE; INFORMATION; CONTAINER; MULTI; USER; SYSTEM; CLIENT; SERVE; NETWORK

Derwent Class: T01

International Patent Class (Main): **G06F-017/30**

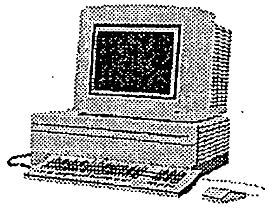
International Patent Class (Additional): **G06F-003/14**

File Segment: EPI

EIC2100

Search Results

Feedback Form (Optional)



Scientific & Technical Information Center

The search results generated for your recent request are attached. If you have any questions or comments (compliments or complaints) about the scope or the results of the search, please contact ***the EIC searcher*** who conducted the search ***or contact***:

Anne Hendrickson, Team Leader, 571-272-3490, RND 4B28
or Carol Wong, Librarian, 571-272-3513, RND 4B28

Voluntary Results Feedback Form

- *I am an examiner in Workgroup:* □ *Example: 2170*
- *Relevant prior art* ***found****, search results used as follows:*

  □ 102 rejection

  □ 103 rejection

  □ Cited as being of interest.

  □ Helped examiner better understand the invention.

  □ Helped examiner better understand the state of the art in their technology.

  Types of relevant prior art found:

  □ Foreign Patent(s)

  □ Non-Patent Literature
  (journal articles, conference proceedings, new product announcements etc.)

- *Relevant prior art* ***not found:***

  □ Results verified the lack of relevant prior art (helped determine patentability).

  □ Search results were not useful in determining patentability or understanding the invention.

Other Comments:

Drop off or send completed forms to **STIC/EIC2100 RND 4B28.**

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 01/10/2005

Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

EXAMINER	
NGUYEN, CAM LINH T	

ART UNIT	PAPER NUMBER
2161	

DATE MAILED: 01/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Interview Summary	**Application No.** 09/284,113	**Applicant(s)** DE ANGELO, MICHAEL
	Examiner CamLinh Nguyen	**Art Unit** 2161

All participants (applicant, applicant's representative, PTO personnel):

(1) *CamLinh Nguyen*. (3)_____.

(2) *Dorian Cartwright (Reg.53,853)*. (4)_____.

Date of Interview: *29 December 2004*.

Type: a)☒ Telephonic b)☐ Video Conference
c)☐ Personal [copy given to: 1)☐ applicant 2)☐ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☒ No.
If Yes, brief description: _____.

Claim(s) discussed: *1 and 9*.

Identification of prior art discussed: *U.S. 6,075,791)*.

Agreement with respect to the claims f)☐ was reached. g)☐ was not reached. h)☒ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Applicant explains more clearly about the invention* and proposed *some amendments to clarify the claim language of the invention.The Applicant agrees to file the amendment to claify the claim language without adding new matter, and the Examiner will consider the amendment. Further action will be made based on the amendment.*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN ONE MONTH FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

ALFORD KINDRED
PRIMARY EXAMINER

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

____
Examiner's signature, if required

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)
In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:

- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 01/26/2005

Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

EXAMINER	
NGUYEN, CAM LINH T	

ART UNIT	PAPER NUMBER
2161	

DATE MAILED: 01/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO 90C (Rev 10/03)

Advisory Action	Application No.	Applicant(s)	
	09/284,113	DE ANGELO, MICHAEL	
	Examiner	Art Unit	
	CamLinh Nguyen	2161	

--The MAILING DATE of this communication appears n the cover sheet with the correspondence address --

THE REPLY FILED 03 January 2005 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE. Therefore, further action by the applicant is required to avoid abandonment of this application. A proper reply to a final rejection under 37 CFR 1.113 may only be either: (1) a timely filed amendment which places the application in condition for allowance; (2) a timely filed Notice of Appeal (with appeal fee); or (3) a timely filed Request for Continued Examination (RCE) in compliance with 37 CFR 1.114.

PERIOD FOR REPLY [check either a) or b)]

a) ☒ The period for reply expires 3 months from the mailing date of the final rejection.

b) □ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.
ONLY CHECK THIS BOX WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

1.□ A Notice of Appeal was filed on _____. Appellant's Brief must be filed within the period set forth in 37 CFR 1.192(a), or any extension thereof (37 CFR 1.191(d)), to avoid dismissal of the appeal.

2.☒ The proposed amendment(s) will not be entered because:

(a) ☒ they raise new issues that would require further consideration and/or search (see NOTE below);

(b) □ they raise the issue of new matter (see Note below);

(c) □ they are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or

(d) □ they present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: *See Continuation Sheet*.

3.□ Applicant's reply has overcome the following rejection(s): _____.

4.□ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).

5.□ The a)□ affidavit, b)□ exhibit, or c)□ request for reconsideration has been considered but does NOT place the application in condition for allowance because: _____.

6.□ The affidavit or exhibit will NOT be considered because it is not directed SOLELY to issues which were newly raised by the Examiner in the final rejection.

7.□ For purposes of Appeal, the proposed amendment(s) a)□ will not be entered or b)□ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.

The status of the claim(s) is (or will be) as follows:

Claim(s) allowed: _____.

Claim(s) objected to: _____.

Claim(s) rejected: _____.

Claim(s) withdrawn from consideration: _____.

8.□ The drawing correction filed on _____ is a)□ approved or b)□ disapproved by the Examiner.

9.□ Note the attached Information Disclosure Statement(s)(PTO-1449) Paper No(s). _____.

10.□ Other: _____

Continuation of 2. NOTE: The amended limitation "an active register for identifying times...with other containers, processes, systems or gateways" to independent claims 1 and 37 has changed the scope of the claims that need further search and reconsideration.

UYEN LE
PRIMARY EXAMINER

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US

CONFIRMATION NO. 1910

OC000000015091147

OC000000015091147

Fish & Richardson
500 Arguello Street
Sute 500
Redwood City, CA 94063

Date Mailed: 02/02/2005

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/03/2004.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

ROSALIND A BALL
2100 (571) 272-3566

OFFICE COPY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US

CONFIRMATION NO. 1910

OC000000015091115

OC000000015091115

Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

Date Mailed: 02/02/2005

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/03/2004.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

ROSALIND A BALL
2100 (571) 272-3566

OFFICE COPY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(c) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US

CONFIRMATION NO. 1910

OC000000015091115

OC000000015091115

Michael De Angelo
Information Equity Corporation
100 South Sunrise Boulevard, Suite 470
Palm Springs, CA 92262

Date Mailed: 02/02/2005

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/03/2004.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

R Ball
ROSALIND A BALL
2100 (571) 272-3566

FORMER ATTORNEY/AGENT COPY

Organization TC2100 Bldg./Room PK2
UNITED STATES PATENT AND TRADEMARK OFFICE
Washington, D.C. 20231
If Undeliverable Return In Ten Days

Official Business
Penalty For Private Use, $300

AN EQUAL OPPORTUNITY EMPLOYER

Return to Sender
☒ No Longer at this Address
□ Recipient Refused Delivery
□ Forwarding has Expired
□ Recipient Unknown
□ Insufficient Address

~~Suite 470~~

[redacted]

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

7590 03/07/2005

Fish & Richardson
500 Arguello Street
Sute 500
Redwood City, CA 94063

EXAMINER	
NGUYEN, CAM LINH T	
ART UNIT	**PAPER NUMBER**
2161	

DATE MAILED: 03/07/2005

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

TITLE OF INVENTION: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$0	$1400	06/07/2005

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE REFLECTS A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE APPLIED IN THIS APPLICATION. THE PTOL-85B (OR AN EQUIVALENT) MUST BE RETURNED WITHIN THIS PERIOD EVEN IF NO FEE IS DUE OR THE APPLICATION WILL BE REGARDED AS ABANDONED.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL should be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). Even if the fee(s) have already been paid, Part B - Fee(s) Transmittal should be completed and returned. If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or Fax (703) 746-4000

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

7590 03/07/2005

Fish & Richardson
500 Arguello Street
Sute 500
Redwood City, CA 94063

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (703) 746-4000, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

TITLE OF INVENTION: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$0	$1400	06/07/2005

EXAMINER	ART UNIT	CLASS-SUBCLASS
NGUYEN, CAM LINH T	2161	707-001000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

□ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

□ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 ______________________

2 ______________________

3 ______________________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : □ Individual □ Corporation or other private group entity □ Government

4a. The following fee(s) are enclosed:

□ Issue Fee

□ Publication Fee (No small entity discount permitted)

□ Advance Order - # of Copies ______________

4b. Payment of Fee(s):

□ A check in the amount of the fee(s) is enclosed.

□ Payment by credit card. Form PTO-2038 is attached.

□ The Director is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number ______________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

□ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

□ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.
NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ______________________ Date ______________________

Typed or printed name ______________________ Registration No. ______________________

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

JN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 03/07/2005

Fish & Richardson
500 Arguello Street
Sute 500
Redwood City, CA 94063

EXAMINER	
NGUYEN, CAM LINH T	
ART UNIT	PAPER NUMBER
2161	

DATE MAILED: 03/07/2005

Determination of Patent Term Extension under 35 U.S.C. 154 (b)

(application filed after June 7, 1995 but prior to May 29, 2000)

The Patent Term Extension is 0 day(s). Any patent to issue from the above-identified application will include an indication of the 0 day extension on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Extension is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571) 272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (703) 305-8283.

Notice of Allowability	**Application No.** 09/284,113	**Applicant(s)** DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	**Art Unit** 2161	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--
All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *amendment filed 01/03/2005*.

2. ☒ The allowed claim(s) is/are *1 – 7, 10, 15 – 19, 37 – 39* .

3. ☒ The drawings filed on *07 April 1999* are accepted by the Examiner.

4. □ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) □ All b) □ Some* c) □ None of the:
   1. □ Certified copies of the priority documents have been received.
   2. □ Certified copies of the priority documents have been received in Application No. _____ .
   3. □ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. □ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. □ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) □ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) □ hereto or 2) □ to Paper No./Mail Date _____.
   (b) □ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. □ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)
1. □ Notice of References Cited (PTO-892)
2. □ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. □ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. □ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. □ Notice of Informal Patent Application (PTO-152)
6. □ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. □ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. □ Other _____.

ALFORD KINDRED
PRIMARY EXAMINER

DETAILED ACTION

Response to Amendment

1. This Office Action is responsive to Amendment papers filed 01/03/2005. Consequently, Advisory Action mailed on 01/26/2005 is withdrawn.

2. Applicant's amendments to the specification are acknowledged. Consequently, the objection to the Specification is withdrawn and the specification has amended to include a new abstract.

3. Applicant's amendments to the claims are acknowledged. Consequently, claims 1, 10, 37 are amended; claims 38 – 39 are added to include limitations of claims 1 and 10 and 37 and 10 respectively. Claims 8 – 9, 11 – 14, 20 – 36 are cancelled. Claims 1 – 7, 10, 15 – 19, 37 – 39 are pending in this application.

4. Claims 1 – 7, 10, 15 – 19, 37 – 39 are renumbered as 1 – 16 respectively.

Allowable Subject Matter

5. Claims 1 – 7, 10, 15 – 19, 37 – 39 are allowed.

6. The following is an examiner's statement of reasons for allowance: in independent claims 1 and 37, an apparatus including a plurality of containers, each container comprising an information element, plurality of registers wherein the plurality of registers including an active time register, passive time register, neutral time register, in conjunction with the other claim limitations, and in independent claims 38 – 39 an apparatus including a plurality of containers,

each container comprising an information element, plurality of registers wherein the plurality of registers including a first register for storing container ID value, an acquire register for controlling whether the container adds a register from other containers or adds a container from other containers when interacting with them, taken with the other limitations of the claim, were not disclosed by, would not have been obvious over, nor otherwise fairly disclosed by the prior art of record.

7. The dependent claims, being further limiting, definite and fully enabled by the Specification, are also allowed.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CamLinh Nguyen whose telephone number is (571) 272 - 4024. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Safet Metjahic can be reached on (571) 272 - 4023. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Nguyen, Cam-Linh

Art Unit 2161

Notice of References Cited	Application/Control No. 09/284,113	Applicant(s)/Patent Under Reexamination DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	Art Unit 2161	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,173,280 B1	01-2001	Ramkumar et al.	707/6
	B	US-6,154,782 A	11-2000	Kawaguchi et al.	709/239
	C	US-6,198,738 B1	03-2001	Chang et al.	370/352
	D	US-5,815,665 A	09-1998	Teper et al.	709/229
	E	US-6,075,791 A	06-2000	Chiussi et al.	370/412
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Issue Classification	**Application No.** 09/284,113	**Applicant(s)** DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	**Art Unit** 2161	

ISSUE CLASSIFICATION

ORIGINAL						CROSS REFERENCE(S)							
CLASS				SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)							
707				100	707	203							
INTERNATIONAL CLASSIFICATION													
G	0	6	F	17/30									
				/									
				/									
				/									
				/									

Nguyen Camlinh 2/23/05
(Assistant Examiner) (Date)

(Legal Instruments Examiner) (Date)

ALFORD KINDRED
PRIMARY EXAMINER
2/24/05
(Primary Examiner) (Date)

Total Claims Allowed: 16

O.G. Print Claim(s)	O.G. Print Fig.
1	2A

□ **Claims renumbered in the same order as presented by applicant** □ CPA □ T.D. □ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1		31		61		91		121		151		181
2	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65		95		125		155		185
6	6		36		66		96		126		156		186
7	7	14	37		67		97		127		157		187
	8	15	38		68		98		128		158		188
	9	16	39		69		99		129		159		189
8	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
9	15		45		75		105		135		165		195
10	16		46		76		106		136		166		196
11	17		47		77		107		137		167		197
12	18		48		78		108		138		168		198
13	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

## Search Notes	Application No. 09/284,113	Applicant(s) DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	Art Unit 2161	

SEARCHED

Class	Subclass	Date	Examiner
707	6	10/25/2004	LINH
707	all	12/30/2004	LINH

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST search	10/25/2004	LINH
EAST search	12/30/2004	LINH
NPL search	1/4/2005	LINH

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner
707	**100**	2/23/2005	**LINH**
707	203	2/23/2005	LINH

Index of Claims

Application No.	Applicant(s)
09/284,113	DE ANGELO, MICHAEL
Examiner	**Art Unit**
CamLinh Nguyen	2161

√	**Rejected**	–	**(Through numeral) Cancelled**	**N**	**Non-Elected**	**A**	**Appeal**
=	**Allowed**	÷	**Restricted**	**I**	**Interference**	**O**	**Objected**

Claim		Date								
Final	Original	10/26/04	2/23/05							
1	1	v	=							
2	2	v	=							
3	3	v	=							
4	4	v	=							
5	5	v	=							
6	6	v	=							
7	7	v	=							
	8									
	9	v								
8	10	v	=							
	11									
	12									
	13									
	14	v								
9	15	v	=							
10	16	v	=							
11	17	v	=							
12	18	v	=							
13	19	v	=							
	20									
	21									
	22									
	23									
	24									
	25									
	26									
	27									
	28									
	29									
	30									
	31									
	32									
	33									
	34									
	35									
	36									
14	37	v	=							
15	38		=							
16	39		=							
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	50									
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIBDATASHEET

Bib Data Sheet

CONFIRMATION NO. 1910

SERIAL NUMBER 09/284,113	FILING DATE 04/07/1999 RULE	CLASS 707	GROUP ART UNIT 2161	ATTORNEY DOCKET NO. 3726-US

APPLICANTS

MICHAEL DE ANGELO, SANTA BARBARA, CA;

** CONTINUING DATA ************************* Yes LN

This application is a 371 of PCT/US99/01988 01/28/1999
which claims benefit of 60/073,209 01/30/1998

** FOREIGN APPLICATIONS ******************** None LN

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 04/12/2000

Foreign Priority claimed □ yes ☑ no 35 USC 119 (a-d) conditions met □ yes ☑ no □ Met after Allowance Verified and Acknowledged ______ Examiner's Signature LN Initials	STATE OR COUNTRY CA	SHEETS DRAWING 30	TOTAL CLAIMS ~~28~~ 16	INDEPENDENT CLAIMS 2

ADDRESS
Fish & Richardson
500 Arguello Street
Sute 500
Redwood City , CA
94063

TITLE
SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

FILING FEE RECEIVED 669	FEES: Authority has been given in Paper No. _________ to charge/credit DEPOSIT ACCOUNT No. _________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other _______________ □ Credit

Sheet 1 of 1

FORM PTO-1449 (REV. 6-89) U.S. DEPARTMENT OF COMMERCE Patent and Trademark Office	Attorney's Docket No. 3726	Serial No. 09/284,113
INFORMATION DISCLOSURE CITATION	Applicant Michael De Angelo	
(Use several sheets if necessary)	Filing Date April 7, 1999	Group Art Unit 2171

U.S. PATENT DOCUMENTS

Examiner Initial		Document Number	Date	Name	Class	Subclass	Filing Date If Appropriate
LN	C	5,664,208	09/02/97	Pavley et al.	395	777	May 16, 1995

FOREIGN PATENT DOCUMENTS

		Document Number	Date	Country	Class	Subclass	Translation	
							Yes	No
LN	D	WO 98 02831	01/22/98	PCT	G06F	17/30		

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER DATE CONSIDERED 1/30/03

EXAMINER: Initial if references considered, whether or not citation is in conformance with MPEP § 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Sheet 1 of 1

FORM PTO-1449
(REV. 6-89)

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

INFORMATION DISCLOSURE CITATION

(Use several sheets if necessary)

Attorney's Docket No. 3726

Serial No. 09/284,118

Applicant Michael De Angelo

Filing Date April 7, 1999

Group Art Unit Unknown

U.S. PATENT DOCUMENTS

Examiner Initial		Document Number							Date	Name	Class	Subclass	Filing Date If Appropriate
LN	A	5	7	6	8	5	1	0	06/16/98	Gish	395	200.33	07/01/96
LN	B	5	8	4	8	2	4	6	12/08/98	Gish	395	200.58	07/01/96

FOREIGN PATENT DOCUMENTS

		Document Number	Date	Country	Class	Subclass	Translation	
							Yes	No

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER Nguyen Cam Linh

DATE CONSIDERED 1/30/03

EXAMINER: Initial if references considered, whether or not citation is in conformance with MPEP § 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

● PRINTER RUSH ●
(PTO ASSISTANCE)

Application :	09/284113	**Examiner :**	Kindred	**GAU :**	2161
From:	PAP	**Location:**	[✓ **IDC**] **FMF FDC**	**Date:**	5/18/05

Tracking #: 06086395 Week Date: 3/14/05

DOC CODE	DOC DATE	MISCELLANEOUS
□ 1449		□ Continuing Data
□ IDS		□ Foreign Priority
☑ CLM	1/3/2005	□ Document Legibility
☑ IIFW	3/7/2005	□ Fees
□ SRFW		□ Other
□ DRW		
□ OATH		
□ 312		
□ SPEC		

[RUSH] **MESSAGE:** Original claims 1 through 7 and renumbered claims 9 through 13 (original claims 15 through 19) depend on renumbered claim 14 (original claim 37).

Thank you

[XRUSH] **RESPONSE:**

INITIALS:

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.
REV 10/04

PART B – FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or Fax (703) 746-4000

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 4 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Legibly mark-up with any corrections or use Block 1)

26181 7590 03/07/2005

FISH & RICHARDSON P.C.
500 ARGUELLO STREET, SUITE 500
REDWOOD CITY, CALIFORNIA 94063

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO, on the date indicated below.

Emma Durrell	(Depositor's name)
Emma Durrell	(Signature)
06/07/2005	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	Michael DeAngelo	17776-002US1	1910

TITLE OF INVENTION: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$700	$0	$700	06/07/2005

EXAMINER	ART UNIT	CLASS-SUBCLASS
NGUYEN, CAM LINH T.	2161	707-001000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

[ ] Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

[ ] "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1. **Fish & Richardson P.C.**
2. ________
3. ________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. Inclusion of assignee data is only appropriate when an assignment has been previously submitted to the USPTO or is being submitted under separate cover. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: **PATTERN INTELLIGENCE, INC.**

(B) RESIDENCE (CITY and STATE OR COUNTRY): **Palm Springs, California**

Please check the appropriate assignee category or categories (will not be printed on the patent): [ ] individual [X] corporation or other private group entity [ ] government

4a. The following fee(s) are enclosed:
[X] Issue Fee
[ ] Publication Fee (No small entity discount permitted)
[X] Advance Order - # of Copies 1

4b. Payment of Fee(s):
[X] A check in the amount of the fee(s) is enclosed.
[ ] Payment by credit card. Form PTO-2038 is attached.
[X] The Director is hereby authorized to charge the required fee(s), or credit any overpayment, to Deposit Account Number 06-1050 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
[X].a. Applicant claims SMALL ENTITY status. See 37 CFR 1.2.7.
[ ]b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.
NOTE: The issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant, a registered agent or; or the assignee or other party in interest as shown by the records of the Untied States Patent and Trademark Office.

(Authorized Signature) Dorian Cartwright

(Date) June 7, 2005

Typed or Printed Name Dorian Cartwright

Registration No. 53,853

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMIT THIS FORM WITH FEE(S)

SUBSTITUTE PTOL-85 (Rev. 12/04) Approved for use through 04/30/2007. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Attorney's Docket No.: 17776-002US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	DeAngelo, Michael	Art Unit :	2161
Serial No.	:	09/284,113	Examiner :	Cam Linh T. Nguyen
Filed	:	April 7, 1999	Confirmation No.:	1910
			Notice of Allowance Date:	March 7, 2005

Title : SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

MAIL STOP ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NOTICE OF ALLOWANCE

In response to the Notice of Allowance mailed March 7, 2005, enclosed are a completed issue fee transmittal form PTOL-85b, a check for $703 for the required fee, including patent copies, a Recordation Form Cover Sheet, and an executed Assignment, and a separate check for $40 for the recordal fee for the above-captioned patent application

Please apply any additional charges or credits to our Deposit Account No. 06-1050.

Respectfully submitted,

Date: 6/7/05

Dorian Cartwright
Dorian Cartwright
Reg. No. 53,853

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

50281375.doc

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

June 7, 2005
Date of Deposit

Emma Durrell
Emma Durrell

● PRINTER RUSH ●

(PTO ASSISTANCE)

Application : 09/284113 Examiner : Kindred GAU : 2161

From: PAP Location: [✓ IDC] FMF FDC Date: 5/18/05

Tracking #: 06086395 Week Date: 3/14/05

DOC CODE	DOC DATE	MISCELLANEOUS
□ 1449		□ Continuing Data
□ IDS		□ Foreign Priority
☑ CLM	1/3/2005	□ Document Legibility
☑ IIFW	3/7/2005	□ Fees
□ SRFW		□ Other
□ DRW		
□ OATH		
□ 312		
□ SPEC		

[RUSH] **MESSAGE:** Original claims 1 through 7 and renumbered claims 9 through 13 (original claims 15 through 19) depend on renumbered claim 14 (original claim 37).

Thank you

[XRUSH] **RESPONSE:** I have reviewed and the claim 2-7, 10; 15-19 (original) depend on either 1 or 37, Therefore, the way that I renumbered was correct same order as in the inclosed Issue Classification

INITIALS: LN

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

Issue Classification	Application No. 09/284,113	Applicant(s) DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	Art Unit 2161	

ISSUE CLASSIFICATION

ORIGINAL						CROSS REFERENCE(S)							
CLASS				SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)							
707				100	707	203							
INTERNATIONAL CLASSIFICATION													
G	0	6	F	17/30									
				/									
				/									
				/									
				/									

Nguyen Camlinh 2/23/05 (Assistant Examiner) (Date)	ALFORD KINDRED PRIMARY EXAMINER 2/24/05 (Primary Examiner) (Date)	Total Claims Allowed: 16
(Legal Instruments Examiner) (Date)		O.G. Print Claim(s): 1 — O.G. Print Fig.: 2A

□ Claims renumbered in the same order as presented by applicant								□ CPA		□ T.D.		□ R.1.47	
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1		31		61		91		121		151		181
2	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65		95		125		155		185
6	6		36		66		96		126		156		186
7	7	14	37		67		97		127		157		187
	8	15	38		68		98		128		158		188
	9	16	39		69		99		129		159		189
8	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
9	15		45		75		105		135		165		195
10	16		46		76		106		136		166		196
11	17		47		77		107		137		167		197
12	18		48		78		108		138		168		198
13	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

GP2161
B1CPW

FISH & RICHARDSON P.C.

PTO/SB/122(06-03)

CHANGE OF CORRESPONDENCE ADDRESS
Application

Address to:
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Application Number	09/284,113
Filing Date	April 7, 1999
First Named Inventor	DeAngelo, Michael
Group Art Unit	2161
Examiner Name	Cam Lihn T. Nguyen
Attorney Docket Number	17776-002US1

Please change the Correspondence Address for the above-identified application to:

☒ **Customer Number:** **26181**

☐ **Firm *or* Individual Name** Dorian Cartwright

Address

Address

City **State** **Zip**

Country United States of America

Telephone (650) 839-5070 **Fax** (650) 839-5071

This form cannot be used to change the data associated with a Customer Number. To change the data associated with an existing Customer Number use "Request for Customer Number Data Change" (PTO/SB/124).

I am the :

☐ Applicant/Inventor.

☐ Assignee of record of the entire interest.
Certificate under 37 CFR 3.73(b) is enclosed.

☒ Attorney or agent of record. Registration Number 53,853

☐ Registered practitioner named in the application transmittal letter in an application without an executed oath or declaration. See 37 CFR 1.33(a)(1). Registration Number 53,853

Typed or Printed Name Dorian Cartwright

Signature Dorian Cartwright

Date 6/23/05 Telephone (650) 839-5070

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of _____ forms are submitted.

● PRINTER RUSH ●

(PTO ASSISTANCE)

2nd Request

Application : 09284113 **Examiner :** Nguyen **GAU :** 2141

From: J Black **Location:** [✓ IDC] FMF FDC **Date:** 7/13/05

Tracking #: 06086395 Week Date: 3/14/05

DOC CODE	DOC DATE	MISCELLANEOUS
□ 1449		□ Continuing Data
□ IDS		□ Foreign Priority
☒ CLM	1/3/05	□ Document Legibility
☒ IIFW	4/20/05	□ Fees
□ SRFW		□ Other
□ DRW		
□ OATH		
□ 312		
□ SPEC		

[RUSH] **MESSAGE:** Original claims 1-7 and renumbered claims 9-13 (originals 15-19) depend upon higher claim, renumbered claim 14 (original claim 37).

Please resolve

[XRUSH] **RESPONSE:**

INITIALS:

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

● PRINTER RUSH ●

(PTO ASSISTANCE)

2nd Request

Application : 09284113 **Examiner :** Nguyen **GAU :** 2141

From: JBlaan **Location:** [✓ IDC] FMF FDC **Date:** 7/13/05

Tracking #: 06086395 Week Date: 3/14/05

DOC CODE	DOC DATE	MISCELLANEOUS
□ 1449		□ Continuing Data
□ IDS		□ Foreign Priority
☒ CLM	1/3/05	□ Document Legibility
☒ IIFW	4/20/05	□ Fees
□ SRFW		□ Other
□ DRW		
□ OATH		
□ 312		
□ SPEC		

[RUSH] **MESSAGE:** Original claims 1-7 and renumbered claims 9-13 (originals 15-19) depend upon higher claim renumbered claim 14 (original claim 37).

Please resolve

[XRUSH] **RESPONSE:** A) Original independent claim 37 is renumbered as independent claim 14

B) Original dependent claims 15-19 are renumbered as dependent claims 9-13 which depend on independent claim 1

C) Original independent claim 1 is renumbered as independent claim 1

D) Claim 9 was canceled.

INITIALS: LN

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

Issue Classification	Application No. 09/284,113	Applicant(s) DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	Art Unit 2161	

ISSUE CLASSIFICATION

ORIGINAL					CROSS REFERENCE(S)							
CLASS				SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)						
707				100	707	203						
INTERNATIONAL CLASSIFICATION												
G	0	6	F	17/30								
				/								
				/								
				/								
				/								

Nguyen CamLinh 2/23/05 (Assistant Examiner) (Date)	ALFORD KINDRED PRIMARY EXAMINER 2/24/05 (Primary Examiner) (Date)	Total Claims Allowed: 16
(Legal Instruments Examiner) (Date)		O.G. Print Claim(s): 1 O.G. Print Fig.: 2A

□ Claims renumbered in the same order as presented by applicant □ CPA □ T.D. □ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1		31		61		91		121		151		181
2	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65		95		125		155		185
6	6		36		66		96		126		156		186
7	7	14	37		67		97		127		157		187
	8	15	38		68		98		128		158		188
	9	16	39		69		99		129		159		189
8	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
9	15		45		75		105		135		165		195
10	16		46		76		106		136		166		196
11	17		47		77		107		137		167		197
12	18		48		78		108		138		168		198
13	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

● PRINTER RUSH ●

(PTO ASSISTANCE)

3rd Request

Application : 09284113 Examiner : Nguyen GAU : 2161

From: J. Black Location: [✓ IDC] FMF FDC Date: 9/17/05

Tracking #: 06086395 Week Date: 3/14/05

DOC CODE	DOC DATE	MISCELLANEOUS
□ 1449		□ Continuing Data
□ IDS		□ Foreign Priority
☒ CLM	1/3/05	□ Document Legibility
☒ IIFW	6/10/05	□ Fees
□ SRFW		□ Other
□ DRW		
□ OATH		
□ 312		
□ SPEC		

[RUSH] **MESSAGE:** Renumbered claims 2-7 and 9-13 (originals 2-7 and 15-19) depend on renumbered claim 14 (original claim 37).

Please resolve.

[XRUSH] **RESPONSE:**

INITIALS:

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	3726-US	1910

7590 09/23/2005

Fish & Richardson
500 Arguello Street
Sutc 500
Redwood City, CA 94063

EXAMINER	
NGUYEN, CAM LINH T	
ART UNIT	**PAPER NUMBER**
2161	

DATE MAILED: 09/23/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

NOTICE OF DRAWING INCONSISTENCY WITH SPECIFICATION

The drawings filed 04 – 07 – 1999 have been received. However, an inconsistency exists between the drawings and the Brief Description of the Drawings in the specification.

Figure ______ is listed in the Brief Description of the Drawings in the specification but not contained in the Drawings.

Figure 3B is contained in the Drawings but not listed in the Brief Description of the Drawings in the specification.

Applicant is required to correct the above-noted inconsistency within a time period of **ONE MONTH or THIRTY (30) DAYS, whichever is longer,** from the mailing date of this Notice, or within the time remaining in the time period set forth in the Notice of Allowability (Form PTOL-37) to file corrected drawings, whichever is longer. **NO EXTENSION OF THIS TIME PERIOD MAY BE GRANTED UNDER EITHER 37 CFR 1.136 (a) OR (b)**

Failure to correct the above noted inconsistency will result in **abandonment** of the application.

The file will be held in the Publishing Division to await the correction of the inconsistency.

Return Corrected Drawings/Specification to:
Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450

Alexandria, VA 22313-1450

Office of Patent Publication/Publishing Division
Customer Service: 703-308-6789
1-888-786-0101

FORM PTO-1631 (REV. 10-03)

● PRINTER RUSH ●

(PTO ASSISTANCE)

3rd Request

Application : 09284113 **Examiner :** Nguyen **GAU :** 2161

From: J. Black **Location:** [✓ IDC] FMF FDC **Date:** 9/17/05

Tracking #: 06086395 Week Date: 3/14/05

DOC CODE	DOC DATE	MISCELLANEOUS
□ 1449		□ Continuing Data
□ IDS		□ Foreign Priority
☒ CLM	1/3/05	□ Document Legibility
☒ IIFW	6/10/05	□ Fees
□ SRFW		□ Other
□ DRW		
□ OATH		
□ 312		
□ SPEC		

[RUSH] **MESSAGE:** Renumbered claims 2-7 and 9-13 (Originals 2-7 and 15-19) depend on renumbured claim 14 (original claim 37).

Please resolve.

[XRUSH] **RESPONSE:** After discuss with lawyer Pinchus in the SPRE Shop please correct the claims as follow:

* Original independent claim 37 is Reinumbered as 2

* Original dependent claim 2 is Renumbered as 14

* see the issue classification

INITIALS: LN

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

Issue Classification	Application No.	Applicant(s)
	09/284,113	DE ANGELO, MICHAEL
	Examiner	Art Unit
	CamLinh Nguyen	2161

ISSUE CLASSIFICATION

ORIGINAL						CROSS REFERENCE(S)						
CLASS	SUBCLASS				CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)						
707	100				707	203						
INTERNATIONAL CLASSIFICATION												
G	0	6	F	17/30								
				/								
				/								
				/								
				/								

Nguyen CamLinh 9/26/05
(Assistant Examiner) (Date)

(Legal Instruments Examiner) (Date)

Frantz Coby
FRANTZ COBY
PRIMARY EXAMINER
(Primary Examiner) (Date)

Total Claims Allowed: 14

O.G. Print Claim(s)	O.G. Print Fig.
1	2A

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1		31		61		91		121		151		181
14	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65		95		125		155		185
6	6		36		66		96		126		156		186
7	7	2	37		67		97		127		157		187
	8	15	38		68		98		128		158		188
	9	16	39		69		99		129		159		189
8	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
9	15		45		75		105		135		165		195
10	16		46		76		106		136		166		196
11	17		47		77		107		137		167		197
12	18		48		78		108		138		168		198
13	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

Issue Classification	Application No. 09/284,113	Applicant(s) DE ANGELO, MICHAEL	
	Examiner CamLinh Nguyen	Art Unit 2161	

ISSUE CLASSIFICATION

ORIGINAL						CROSS REFERENCE(S)							
CLASS				SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)							
707				100	707	203							
INTERNATIONAL CLASSIFICATION													
G	0	6	F	17/30									
				/									
				/									
				/									
				/									

Nguyen CamLinh 9/26/05
(Assistant Examiner) (Date)

(Legal Instruments Examiner) (Date)

Frantz Coby
FRANTZ COBY
PRIMARY EXAMINER
(Primary Examiner) (Date)

Total Claims Allowed: ~~14~~ 16

O.G. Print Claim(s)	O.G. Print Fig.
1	2A

□ Claims renumbered in the same order as presented by applicant								□ CPA		□ T.D.		□ R.1.47	
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1		31		61		91		121		151		181
14	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65		95		125		155		185
6	6		36		66		96		126		156		186
7	7	2	37		67		97		127		157		187
	8	15	38		68		98		128		158		188
	9	16	39		69		99		129		159		189
8	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
9	15		45		75		105		135		165		195
10	16		46		76		106		136		166		196
11	17		47		77		107		137		167		197
12	18		48		78		108		138		168		198
13	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

Attorney's Docket No.: 17776-002US1

OFFICIAL COMMUNICATION FACSIMILE:

OFFICIAL FAX NO: (571) 273-8300

Number of pages including this page 5

Applicant	: DeAngelo, Michael	Art Unit	: 2161
Serial No.	: 09/284,113	Examiner	: Cam Lihn T. Nguyen
Filed	: April 7, 1999		

Title : System and Method for Creating and Manipulating Information Containers with Dynamic Registers

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attached to this facsimile communication cover sheet is a Response to Notice of Drawing Inconsistency and copy of the Notice of Drawing Inconsistency with Specification, faxed this 24th day of October, 2005, to the United States Patent and Trademark Office.

Respectfully submitted,

Date: October 24, 2005

for Carl A. Kukkonen, III
Reg. No. 42,773

WILLIAM E. HUNTER
REG. NO 47,671

Fish & Richardson P.C.
12390 El Camino Real
San Diego, California 92130
Telephone: (858) 678-5070
Fax: (858) 678-5099

10564047.doc

NOTE: This facsimile is intended for the addressee only and may contain privileged or confidential information. If you have received this facsimile in error, please immediately call us collect at (858) 678-5070 to arrange for its return. Thank you.

Attorney's Docket No.: 17776-002US1/

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: DeAngelo, Michael	Art Unit:	2161
Serial No.	: 09/284,113	Examiner:	Cam Lihn T. Nguyen
Filed	: April 7, 1999		
Title	: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS		

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NOTICE OF DRAWING INCONSISTENCY

In response to Notice of Drawing Inconsistency mailed September 23, 2005 (copy enclosed), please amend the application as follows:

Amendments to the Specification begin on page 2 of this paper.

Remarks/Arguments begin on page 3 of this paper.

CERTIFICATE OF TRANSMISSION BY FACSIMILE

I hereby certify that this correspondence is being transmitted by facsimile to the Patent and Trademark Office on the date indicated below.

October 24, 2005
Date of Transmission

Veronica Whalen
Signature

Veronica Whalen
Typed or Printed Name of Person Signing Certificate

UNITED STATES PATENT AND TRADEMARK OFFICE

UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

NOTICE OF DRAWING INCONSISTENCY WITH SPECIFICATION

The drawings filed 04 – 07 – 1999 have been received. However, an inconsistency exists between the drawings and the Brief Description of the Drawings in the specification.

Figure _____ is listed in the Brief Description of the Drawings in the specification but not contained in the Drawings.

Figure 3B is contained in the Drawings but not listed in the Brief Description of the Drawings in the specification.

Applicant is required to correct the above-noted inconsistency within a time period of **ONE MONTH or THIRTY (30) DAYS, whichever is longer,** from the mailing date of this Notice, or within the time remaining in the time period set forth in the Notice of Allowability (Form PTOL-37) to file corrected drawings, whichever is longer. **NO EXTENSION OF THIS TIME PERIOD MAY BE GRANTED UNDER EITHER 37 CFR 1.136 (a) OR (b)**

Failure to correct the above noted inconsistency will result in **abandonment** of the application.

The file will be held in the Publishing Division to await the correction of the inconsistency.

Return Corrected Drawings/Specification to:
Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450

Alexandria, VA 22313-1450

Office of Patent Publication/Publishing Division
Customer Service: 703-308-6789
1-888-786-0101

FORM PTO-1631 (REV. 10-03)

Amendments to the Specification:

Please add a new paragraph after the paragraph beginning at page 9, line 2:

Fig. 3B is a graphical representation for a second embodiment of a container having a plurality of containers nested within that container.

Attorney's Docket No.:17776-002US1

REMARKS

The specification has been amended to add a brief description for Fig. 3B (for support, see, inter alia, specification page 17, lines 3-12). No new matter is added. Hence, the application as amended is now in full condition for issuance.

Applicants ask that this matter now proceed to issuance. Please apply any applicable charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: Oct. 24, 2005

for Carl A. Kukkonen, III
Reg. No. 42,773

WILLIAM E. HUNTER
REG. NO 47,671

Fish & Richardson P.C.
12390 El Camino Real
San Diego, California 92130
(858) 678-5070 telephone
(858) 678-5099 facsimile

10563724.doc

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/284,113	04/07/1999	MICHAEL DE ANGELO	17776-002US1	1910

26181 7590 01/17/2006

FISH & RICHARDSON P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER
NGUYEN, CAM LINH T

ART UNIT	PAPER NUMBER
2161	

DATE MAILED: 01/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Response to Rule 312 Communication	Application No. 09284113	Applicant(s)	
	Examiner	Art Unit	

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address---

1. ☑ The amendment filed on 10-24-05 under 37 CFR 1.312 has been considered, and has been:

a) ☑ entered.

b) □ entered as directed to matters of form not affecting the scope of the invention.

c) □ disapproved because the amendment was filed after the payment of the issue fee.

Any amendment filed after the date the issue fee is paid must be accompanied by a petition under 37 CFR 1.313(c)(1) and the required fee to withdraw the application from issue.

d) □ disapproved. See explanation below.

e) □ entered in part. See explanation below.

a. Marty Willis

Publishing Division

☆ U.S. GOVERNMENT PRINTING OFFICE: 2003-300-168

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIBDATASHEET

Bib Data Sheet

CONFIRMATION NO. 1910

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
09/284,113	04/07/1999 RULE	707	2161	17776-002US1

APPLICANTS

MICHAEL DE ANGELO, SANTA BARBARA, CA;

** CONTINUING DATA *************************

This application is a 371 of PCT/US99/01988 01/28/1999
which claims benefit of 60/073,209 01/30/1998

** FOREIGN APPLICATIONS ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** SMALL ENTITY **
** 04/12/2000

	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed □ yes □ no 35 USC 119 (a-d) conditions met □ yes □ no □ Met after Allowance Verified and Acknowledged ______ Examiner's Signature Initials	CA	30	36	3

ADDRESS
26181
FISH & RICHARDSON P.C.
PO BOX 1022
MINNEAPOLIS , MN
55440-1022

TITLE
SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

FILING FEE RECEIVED 669	FEES: Authority has been given in Paper No. _________ to charge/credit DEPOSIT ACCOUNT No. _________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other ____________

PTO/SB/83 (01-06)
Approved for use through 12/31/2008. OMB 0651-0035
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REQUEST FOR WITHDRAWAL AS ATTORNEY OR AGENT AND CHANGE OF CORRESPONDENCE ADDRESS

Application Number	09/284,113
Filing Date	April 7, 1999
First Named Inventor	Michael De Angelo
Art Unit	2171
Examiner Name	Cam Linh T. Nguyen
Attorney Docket Number	PATT-002/01US

To: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Please withdraw me as attorney or agent for the above identified patent application, and

[ ] all the attorneys/agents of record.

[ ] the attorneys/agents (with registration numbers) listed on the attached paper(s), or

[X] the attorneys/agents associated with Customer Number 23419

NOTE: This box can only be checked when the power of attorney of record in the application is to all the practitioners associated with a customer number.

The reasons for this request are: **Disengaging due to lack of payment.**

CORRESPONDENCE ADDRESS

1. [ ] The correspondence address is NOT affected by this withdrawal.

2. [X] Change the correspondence address and direct all future correspondence to:

[ ] The address associated with Customer Number:

OR

[x] Firm *or* Individual Name	Michael De Angelo, Pattern Intelligence, Inc.		
Address	6796 Giovanetti Road		
City	Forestville	State CA	Zip 95472
Country	U.S.A.		
Telephone	Direct: (760) 799-0379	Email ab9ac99@yahoo.com	

Signature		Registration No. 33,885
Name	William S. Galliani	Telephone No. 650-843-5000
Date	May 8, 2008	

NOTE: Withdrawal is effective when approved rather than when received. Unless there are at least 30 days between approval of withdrawal and the expiration date of a time period for response or possible extension period, the request to withdraw is normally disapproved.

This collection of information is required by 37 CFR 1.36. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	3279720
Application Number:	09284113
International Application Number:	
Confirmation Number:	1910
Title of Invention:	SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS
First Named Inventor/Applicant Name:	MICHAEL DE ANGELO
Customer Number:	26181
Filer:	William S. Galliani/Gina Luna
Filer Authorized By:	William S. Galliani
Attorney Docket Number:	17776-002US1
Receipt Date:	09-MAY-2008
Filing Date:	07-APR-1999
Time Stamp:	12:41:49
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Change of Address	Request_For_Withdrawal.pdf	140180 0caa2297ed7ce8903763a0dc33b3029 8ceab769c	no	1

Warnings:

Information:

Total Files Size (in bytes):	140180

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/284,113	04/07/1999	MICHAEL DE ANGELO	17776-002US1

CONFIRMATION NO. 1910

IMPROPER CPOA LETTER

OC000000030002140

26181
FISH & RICHARDSON P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

Date Mailed: 05/21/2008

NOTICE REGARDING POWER OF ATTORNEY

This is in response to the Power of Attorney filed 05/09/2008. The Power of Attorney in this application is not accepted for the reason(s) listed below:

- The attorney or agent requesting to withdraw has not been appointed attorney of record in the application (i.e., there is no power of attorney). Accordingly, the withdrawal cannot be accepted because the practitioner seeking to withdraw is not of record. Moreover, a patent practitioner cannot withdraw another practitioner of record unless they also withdraw himself or herself, and affirm that they are acting on behalf of the other patent practitioner(s). To revoke the power of attorney given to another patent practitioner, the applicant must comply with 37 CFR 1.36(a).

/atesfai/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/284,113
Patent No.	:	7,010,536
Applicant	:	MICHAEL DE ANGELO
Filing Date	:	04-07-1999
Issue Date	:	03-07-2006
Title	:	SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS
Confirmation No.	:	1910
Docket No.	:	20933.4001
Customer No.	:	34313

Via EFS Web
Attn: Office of Petitions

REQUEST FOR CERTIFICATE OF CORRECTION OF PATENT FOR APPLICANT'S MISTAKE (37 CFR 1.323)

Dear Sir:

1. Attached is Form PTO/SB/44 with at least one copy being suitable for printing

2. Applicant requests a correction of following words which were inadvertently included on the above-referenced patent due to a typographical mistake.

Claim 1, Page 30, Line 26
intersect (typo) should be replaced with **interact** (correct)

Claim 2, Page 30, Line 49
resister (typo) should be replaced with **register** (correct)

3. Applicant believes than an erroneous placement of the word “time” instead of the word “space” (Claim 16, page 32, line 33), and the word "event time" instead of the word "three dimensional space" (Claim 16, page 32, line 37) occurred due to a clerical error related to a cut and paste operation while employing parallel

construction. Specification and argument indicate the logic of the requested correction.

Claim 16, Page 32, Line 33
"time" (incorrect) should be replaced with "**space"** (correct)

Claim 16, page 32, line 37
"event time" (incorrect) should be replaced with "**three dimensional space**" (correct)

Remarks: Applicant indicates that the corrections are supported by syntax, parallel construction, and word use in the claims themselves, especially between Claims 15 and 16, and in the drawings, and in the summary and specification throughout. The correction is further supported by arguments for Claim 16 to the USPTO, January 3, 2005, our attorney argument versus Chiussi as not including "space". Also "space" is already established in Claim 2; therefore the change does not expand because Claim 16, becomes only a combination of "space" as already granted in Claim 1, and "acquire" as already granted in Claim 15 and 16, and shown thought the specification and in Drawings, Figure 4, Registers "Acquire" (123000) and "Active Space" (111000), "Passive Space"(112000), and "Neutral Space" (113000), and described processes in Preferred Embodiment, and as describing processes in Figure 13 B, describing the acquisition of time and "space relevant" containers, and in Figure 13 C, and in Figure 14, "acquire 1404, edit 1405, or create a register 1406 to append or detach registers 120 to those containers" and Figure 16, and in paragraphs describing "an acquire register 123000, enabling the user to search and utilize other registers residing on the network".

4. Applicant believes that the above-referenced corrections do not constitute in the addition of new matter.

5. Fee and Method of Fee Payment for Certificate of Correction:

☐ To our knowledge, no fees are required.

☐ Attached is a check for ________.

☒ Charge Account 15-0665 ($ 100.00).

☒ If any additional fees are due, please charge Account 15-0665.

6 Please forward the Certificate to the undersigned.

Respectfully submitted,

ORRICK, HERRINGTON & SUTCLIFFE LLP

Dated: May 2, 2012

By: /Mark Shean/
Mark J. Shean
Reg. No. 54441

Orrick, Herrington & Sutcliffe LLP
2050 MAIN STREET, SUITE 1100
IRVINE, CA 92614-8255
Tel. 949-567-6700
Fax: 949-567-6710

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

Page 1 of 1

PATENT NO. : 7010536

APPLICATION NO.: 09/284,113

ISSUE DATE: : 03-07-2006

INVENTOR(S) : MICHAEL DE ANGELO

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:
Claim 1, Page 30, Line 26,
between the words “may” and “with” delete the word “intersect", insert therefor the word "interact"
Claim 2, Page 30, Line 49
between the words “passive” and “for” delete the word “resister", insert therefor the word "register"
Claim, 16, Page 32, Line 33
between the words “designating" (from previous line) and “and governing” delete the word “time", insert therefor the word "space"
Claim, 16, Page 32, Line 37
between the words “external-to-the-apparatus" and “,and” delete the word “event time", insert therefor the word "three dimensional space"

MAILING ADDRESS OF SENDER (Please do not use customer number below):
ORRICK, HERRINGTON & SUTCLIFFE LLP
2050 MAIN STREET, SUITE 1100
IRVINE CA 92614-8255

This collection of information is required by 37 CFR 1.322, 1.323, and 1.324. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1.0 hour to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Attention Certificate of Corrections Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/96 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: MICHAEL DE ANGELO

Application No./Patent No.: 7010536 Filed/Issue Date: 03/07/2006

Entitled: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

INCANDESCENT, INC. , a CORPORATION
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☒ the assignee of the entire right, title, and interest in;

2. □ an assignee of less than the entire right, title and interest in
(The extent (by percentage) of its ownership interest is ______%); or

3. □ the assignee of an undivided interest in the entirety of (a complete assignment from one of the joint inventors was made)

in the patent application/patent identified above by virtue of either:

A. □ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel ______, Frame ______, or for which a copy thereof is attached.

OR

B. □ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: ____________________ To: ____________________
The document was recorded in the United States Patent and Trademark Office at
Reel ______, Frame ______, or for which a copy thereof is attached.

2. From: ____________________ To: ____________________
The document was recorded in the United States Patent and Trademark Office at
Reel ______, Frame ______, or for which a copy thereof is attached.

3. From: ____________________ To: ____________________
The document was recorded in the United States Patent and Trademark Office at
Reel ______, Frame ______, or for which a copy thereof is attached.

□ Additional documents in the chain of title are listed on a supplemental sheet(s).

☒ As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (*i.e.*, a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Signature

May 2, 2012
Date

MICHAEL DE ANGELO
Printed or Typed Name

CEO
Title

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/81A (12-08)
Approved for use through 11/30/2011. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT - POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

Patent Number	7010536
Issue Date	07-Mar-2006
First Named Inventor	Michael J. DE ANGELO
Title	SYSTEM AND METHOD FOR CRE/
Attorney Docket Number	20933.4001

I hereby revoke all previous powers of attorney given in the above-identified patent.

□ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s) with respect to the patent identified above, and to transact all business in the United States Patent and Trademark Office connected therewith: 34313

OR

□ I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) with respect to the patent identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

Practitioner(s) Name	Registration Number

Please recognize or change the correspondence address for the above-identified patent to:

☒ The address associated with the above-mentioned Customer Number.

OR

□ The address associated with Customer Number:

OR

□ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the:

□ Inventor, having ownership of the patent.

OR

☒ Patent owner.
Statement under 37 CFR 3.73(b) (Form PTO/SB/96) submitted herewith or filed on ____________.

SIGNATURE of Inventor or Patent Owner

Signature	Michael DeAngelo	Date	MAY 2, 2012
Name	MICHAEL DE ANGELO	Telephone	
Title and Company	CEO, INCANDESCENT, INC		

NOTE: Signatures of all the inventors or patent owners of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

□ *Total of ____________ forms are submitted.

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

United States Patent and Trademark Office

Home | Site Index | Search | Guides | Contacts | eBusiness | eBiz alerts | News | Help

Assignments on the Web > Patent Query

Patent Assignment Abstract of Title

NOTE:Results display only for issued patents and published applications. For pending or abandoned applications please consult USPTO staff.

Total Assignments: 4

Patent #: 7010536 **Issue Dt:** 03/07/2006 **Application #:** 09284113 **Filing Dt:** 04/07/1999

Inventor: MICHAEL DE ANGELO

Title: SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS

Assignment: 1

Reel/Frame: 010517/0456 **Recorded:** 04/07/1999 **Pages:** 2

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: DE ANGELO, MICHAEL **Exec Dt:** 04/05/1999

Assignee: EMATRIX CORPORATION
SUITE C
104 WEST ANAPAMU
SANTA BARBARA, CALIFORNIA 93101

Correspondent: FENWICK & WEST LLP
GREG T. SUEOKA, ESQ.
TWO PALO ALTO SQUARE
PALO ALTO, CA 94306

Assignment: 2

Reel/Frame: 016675/0763 **Recorded:** 06/09/2005 **Pages:** 2

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: EMATRIX CORPORATION **Exec Dt:** 06/06/2005

Assignee: PATTERN INTELLIGENCE, INC.
100 SOUTH SUNRISE WAY
PALM SPRINGS, CALIFORNIA 92264

Correspondent: FISH & RICHARDSON P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

Assignment: 3

Reel/Frame: 022878/0481 **Recorded:** 06/29/2009 **Pages:** 3

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: PATTERN INTELLIGENCE, INC. **Exec Dt:** 06/26/2009

Assignee: INCANDESCENT, INC.
P.O. BOX 15194
SAN RAFAEL, CALIFORNIA 94915

Correspondent: SANJEET K. DUTTA
4 PARK PLAZA
SUITE 1600
IRVINE, CA 92614

Assignment: 4

Reel/Frame: 022886/0273 **Recorded:** 06/29/2009 **Pages:** 3

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: EMATRIX CORPORATION **Exec Dt:** 06/26/2009

Assignee: PATTERN INTELLIGENCE, INC.
P.O. BOX 15194
SAN RAFAEL, CALIFORNIA 94915

Correspondent: SANJEET K. DUTTA
4 PARK PLAZA
SUITE 1600
IRVINE, CA 92614

Search Results as of: 05/02/2012 09:56 PM
If you have any comments or questions concerning the data displayed, contact PRD / Assignments at 571-272-3350. v.2.3.1
Web interface last modified: Jan 26, 2012 v.2.3.1

| .HOME | INDEX| SEARCH | eBUSINESS | CONTACT US | PRIVACY STATEMENT

Electronic Patent Application Fee Transmittal

Application Number:	09284113
Filing Date:	07-Apr-1999
Title of Invention:	SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS
First Named Inventor/Applicant Name:	MICHAEL DE ANGELO
Filer:	Mark Jonathan Shean/Victor Santos
Attorney Docket Number:	17776-002US1

Filed as Large Entity

U.S. National Stage under 35 USC 371 Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Certificate of correction	1811	1	100	100
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
		Total in USD ($)		**100**

Electronic Acknowledgement Receipt

EFS ID:	12691189
Application Number:	09284113
International Application Number:	
Confirmation Number:	1910
Title of Invention:	SYSTEM AND METHOD FOR CREATING AND MANIPULATING INFORMATION CONTAINERS WITH DYNAMIC REGISTERS
First Named Inventor/Applicant Name:	MICHAEL DE ANGELO
Customer Number:	26181
Filer:	Mark Jonathan Shean/Victor Santos
Filer Authorized By:	Mark Jonathan Shean
Attorney Docket Number:	17776-002US1
Receipt Date:	02-MAY-2012
Filing Date:	07-APR-1999
Time Stamp:	22:08:10
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$100
RAM confirmation Number	9157
Deposit Account	150665
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. 1.492 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	transmittal.pdf	273715 f040ad00733b0669b4a6af13f4cb895703998473	no	4
Warnings:					
Information:					
2		POA.PDF	354807 2fcac14f4daa693cc01fb8faeb050ff83950e967	yes	4

Multipart Description/PDF files in .zip description

Document Description	Start	End
Assignee showing of ownership per 37 CFR 3.73(b).	1	1
Power of Attorney	2	4

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	30273 d5fbff59791787a0377fc3d000d88297afed8f18	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	658795	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

SPE RESPONSE FOR CERTIFICATE OF CORRECTION

Paper No.: ________

DATE : 05-11-12

TO SPE OF : ART UNIT 2161

SUBJECT : Request for Certificate of Correction for Appl. No.: 09/284113 Patent No.: 7010536

COCIN mailroom date: 05-02-12

Please respond to this request for a certificate of correction within 7 days.

FOR IFW FILES:

Please review the requested changes/corrections as shown in the **COCIN** document(s) in the IFW application image. No new matter should be introduced nor should the scope or meaning of the claims be changed.

Please complete the response (see below) and forward the completed response to scanning using document code **COCX**.

FOR PAPER FILES:

Please review the requested changes/corrections as shown in the attached certificate of correction. Please complete this form (see below) and forward it with the file to:

Certificates of Correction Branch (CofC)
Randolph Square – 9D10-A
Palm Location 7580

In particular note: ____________________

Certificates of Correction Branch
703-756-1814

Thank You For Your Assistance

The request for issuing the above-identified correction(s) is hereby:

Note your decision on the appropriate box.

□ **Approved**	**All** changes apply.
□ **Approved in Part**	Specify below which changes **do not** apply.
□ **Denied**	State the reasons for denial below.

Comments: ____________________

____

____

____

____

SPE **Art Unit**

SPE RESPONSE FOR CERTIFICATE OF CORRECTION

Paper No.: ________

DATE : 05-11-12

TO SPE OF : ART UNIT 2161

SUBJECT : Request for Certificate of Correction for Appl. No.: 09/284113 Patent No.: 7010536

COCIN mailroom date: 05-02-12

Please respond to this request for a certificate of correction within 7 days.

FOR IFW FILES:

Please review the requested changes/corrections as shown in the **COCIN** document(s) in the IFW application image. No new matter should be introduced nor should the scope or meaning of the claims be changed.

Please complete the response (see below) and forward the completed response to scanning using document code **COCX**.

FOR PAPER FILES:

Please review the requested changes/corrections as shown in the attached certificate of correction. Please complete this form (see below) and forward it with the file to:

Certificates of Correction Branch (CofC)
Randolph Square – 9D10-A
Palm Location 7580

In particular note: ____________________

Certificates of Correction Branch
703-756-1814

Thank You For Your Assistance

The request for issuing the above-identified correction(s) is hereby:
Note your decision on the appropriate box.

☒ **Approved**	**All** changes apply.
□ **Approved in Part**	Specify below which changes **do not** apply.
□ **Denied**	State the reasons for denial below.

Comments: ____________________

/Apu Mofiz/
SPE

2161
Art Unit

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/284,113	04/07/1999	MICHAEL DE ANGELO	17776-002US1

CONFIRMATION NO. 1910

26181
FISH & RICHARDSON P.C. (SV)
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

POWER OF ATTORNEY NOTICE

OC000000054363087

Date Mailed: 05/23/2012

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 05/02/2012.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/ttkim/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/284,113	04/07/1999	MICHAEL DE ANGELO	20933.4001

CONFIRMATION NO. 1910

POA ACCEPTANCE LETTER

OC000000054363108

34313
ORRICK, HERRINGTON & SUTCLIFFE, LLP
IP PROSECUTION DEPARTMENT
2050 Main Street, Suite 1100
IRVINE, CA 92614

Date Mailed: 05/23/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 05/02/2012.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/ttkim/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,010,536 B1
APPLICATION NO. : 09/284113
DATED : March 7, 2006
INVENTOR(S) : Michael De Angelo

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Claim 1, Page 30, Line 26,
between the words “may” and “with” delete the word “intersect”, insert therefor the word --interact--
Claim 2, Page 30, Line 49,
between the words “passive” and “for” delete the word “resister”, insert therefor the word --register--
Claim, 16, Page 32, Line 33,
between the words “designating” (from previous line) and “and governing” delete the word “time”, insert therefor the word --space--
Claim, 16, Page 32, Line 37,
between the words “external-to-the-apparatus” and “, and” delete the word “event time”, insert therefor the word --three dimensional space--

Signed and Sealed this
Twenty-sixth Day of June, 2012

David J. Kappos

David J. Kappos
Director of the United States Patent and Trademark Office

PATENT ASSIGNMENT

Electronic Version v1.1
Stylesheet Version v1.1

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	ASSIGNMENT

CONVEYING PARTY DATA

Name	Execution Date
Incandescent, Inc.	07/11/2012

RECEIVING PARTY DATA

Name:	Evolutionary Intelligence, LLC
Street Address:	c/o Gutride Safier LLP, 835 Douglass Street
City:	San Francisco
State/Country:	CALIFORNIA
Postal Code:	94114

PROPERTY NUMBERS Total: 3

Property Type	Number
Patent Number:	7010536
Patent Number:	7702682
Patent Number:	7873682

CORRESPONDENCE DATA

Fax Number:	4154496469

Correspondence will be sent via US Mail when the fax attempt is unsuccessful.

Phone:	415-789-6390
Email:	todd@gutridesafier.com
Correspondent Name:	Todd Kennedy
Address Line 1:	835 Douglass Street
Address Line 4:	San Francisco, CALIFORNIA 94114

NAME OF SUBMITTER:	Todd Kennedy
	This document serves as an Oath/Declaration (37 CFR 1.63).

Total Attachments: 2
source=patent assignment executed and notarized#page1.tif
source=patent assignment executed and notarized#page2.tif

PATENT ASSIGNMENT

Inventor:	Michael De Angelo
U.S. Patent Nos.:	7,010,536; 7,702,682; 7,873,682
Application Nos.:	09/284,113; 11/280,700; 12/691,425
Filing Dates:	1/28/99; 11/14/05; 1/21/10
Titles:	"System and method for creating and manipulating information containers with dynamic registers"

WHEREAS, Incandescent, Inc., a Delaware corporation having an office and place of business at 100 Pine Street, San Francisco, California 94111 (hereinafter referred to as "ASSIGNOR") is the current assignee of the U.S. Patents listed above (hereinafter referred to as the "PATENTS"); and

WHEREAS, Evolutionary Intelligence, LLC, a Delaware limited liability company having an office and place of business at 100 Pine Street, Suite 500, San Francisco, California 94111 (hereinafter referred to as "ASSIGNEE"), is desirous of acquiring the entire right, title and interest throughout the world in and to the PATENTS and in and to any letters patent that may be granted therefor in the United States and in any and all foreign countries.

NOW, THEREFORE, in exchange for good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged, ASSIGNOR hereby assigns and transfers unto ASSIGNEE, the entire right, title and interest throughout the world in and to the PATENTS, the inventions described therein, and any and all letters patent which may be granted or have been granted, including all potential, existing, and future causes of action and associated past, present and future damages, for said inventions in the United States of America and its territorial possessions and in any and all foreign countries, and in any and all divisions, reissues, continuations, continuations-in-part, and certificates of correction thereof, including the right to file domestic and foreign applications directly in the name of ASSIGNEE and to claim priority rights deriving from the applications leading to the PATENTS, said inventions, PATENTS, and all other letters patent deriving from said invention to be held and enjoyed by ASSIGNEE and its successors and assigns for their use and benefit and of their successors and assigns as fully and entirely as the same would have been held and enjoyed by ASSIGNOR had this assignment not been made.

ASSIGNOR hereby authorizes and requests the Commissioner of Patents and Trademarks to issue all letters patent on said inventions to ASSIGNEE. ASSIGNOR warrants that ASSIGNOR is the rightful owner of the PATENTS and said invention, and that there are no others who could make a claim against the rights being assigned, and that the rights being assigned are subsisting and are not assigned, licensed, or otherwise diluted in any way. ASSIGNOR further agrees to execute and deliver any further papers and do such other acts as may be necessary and proper to vest full title in and to the PATENTS in the ASSIGNEE. ASSIGNOR further agrees to execute all instruments and documents required for the making and prosecution of foreign and domestic applications for letters patent on said inventions, and for litigation regarding said letters patent. ASSIGNOR also covenants that, if there are any disputes, actions, litigations, trials, or any other challenges related to the rights being assigned, then ASSIGNOR shall assist ASSIGNEE to the best of ASSIGNOR's ability.

IN WITNESS WHEREOF, ASSIGNOR has caused these presents to be duly executed in a manner appropriate thereto this 9th day of July, 2012.

ASSIGNOR:
Incandescent, Inc.

By: Michael DeAngelo
Michael De Angelo
CEO

Signed before me this 11th day of July, 2012.

Dave Berghoff
Notary Public

DAVE BERGHOFF
Comm. #[illegible]
Notary Public - California
Riverside County
My Commission Expires
March 18, 2013

ASSIGNEE hereby accepts this assignment.

ASSIGNEE:
Evolutionary Intelligence, LLC

By: Michael DeAngelo
Michael De Angelo
Manager

Signed before me this 11th day of July, 2012.

Dave Berghoff
Notary Public

DAVE BERGHOFF
Comm. #[illegible]
Notary Public - California
Riverside County
My Commission Expires
March 18, 2013

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:12-cv-794	DATE FILED 10/17/2012	U.S. DISTRICT COURT Eastern District of Texas
PLAINTIFF Evolutionary Intelligence, LLC		DEFENDANT Yelp, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:12-cv-783	DATE FILED 10/17/2012	U.S. DISTRICT COURT Eastern District of Texas
PLAINTIFF Evolutionary Intelligence, LLC		DEFENDANT Apple, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:12-cv-784	10/17/2012	Eastern District of Texas

PLAINTIFF	DEFENDANT
Evolutionary Intelligence, LLC	Facebook, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:12-cv-785	DATE FILED 10/17/2012	U.S. DISTRICT COURT Eastern District of Texas
PLAINTIFF Evolutionary Intelligence, LLC		DEFENDANT Foursquare Labs, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:12-cv-787	DATE FILED 10/17/2012	U.S. DISTRICT COURT Eastern District of Texas
PLAINTIFF Evolutionary Intelligence, LLC		DEFENDANT Groupon, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE** **FILING OR DETERMINATION OF AN** **ACTION REGARDING A PATENT OR** **TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:12-cv-789	DATE FILED 10/17/2012	U.S. DISTRICT COURT Eastern District of Texas

PLAINTIFF Evolutionary Intelligence, LLC	DEFENDANT Livingsocial, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:12-cv-790	10/17/2012	Eastern District of Texas

PLAINTIFF	DEFENDANT
Evolutionary Intelligence, LLC	Millennial Media, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:12-cv-791	DATE FILED 10/17/2012	U.S. DISTRICT COURT Eastern District of Texas

PLAINTIFF Evolutionary Intelligence, LLC	DEFENDANT Sprint Nextel Corporation, Sprint Communications Company L.P., Sprint Spectrum, L.P., Sprint Solutions, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:12-cv-792	10/17/2012	Eastern District of Texas

PLAINTIFF	DEFENDANT
Evolutionary Intelligence, LLC	Twitter, Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,010,536	3/7/2006	Evolutionary Intelligence, LLC
2 7,702,682	4/20/2010	Evolutionary Intelligence, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	**DATE OF PATENT OR TRADEMARK**	**HOLDER OF PATENT OR TRADEMARK**
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy