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.

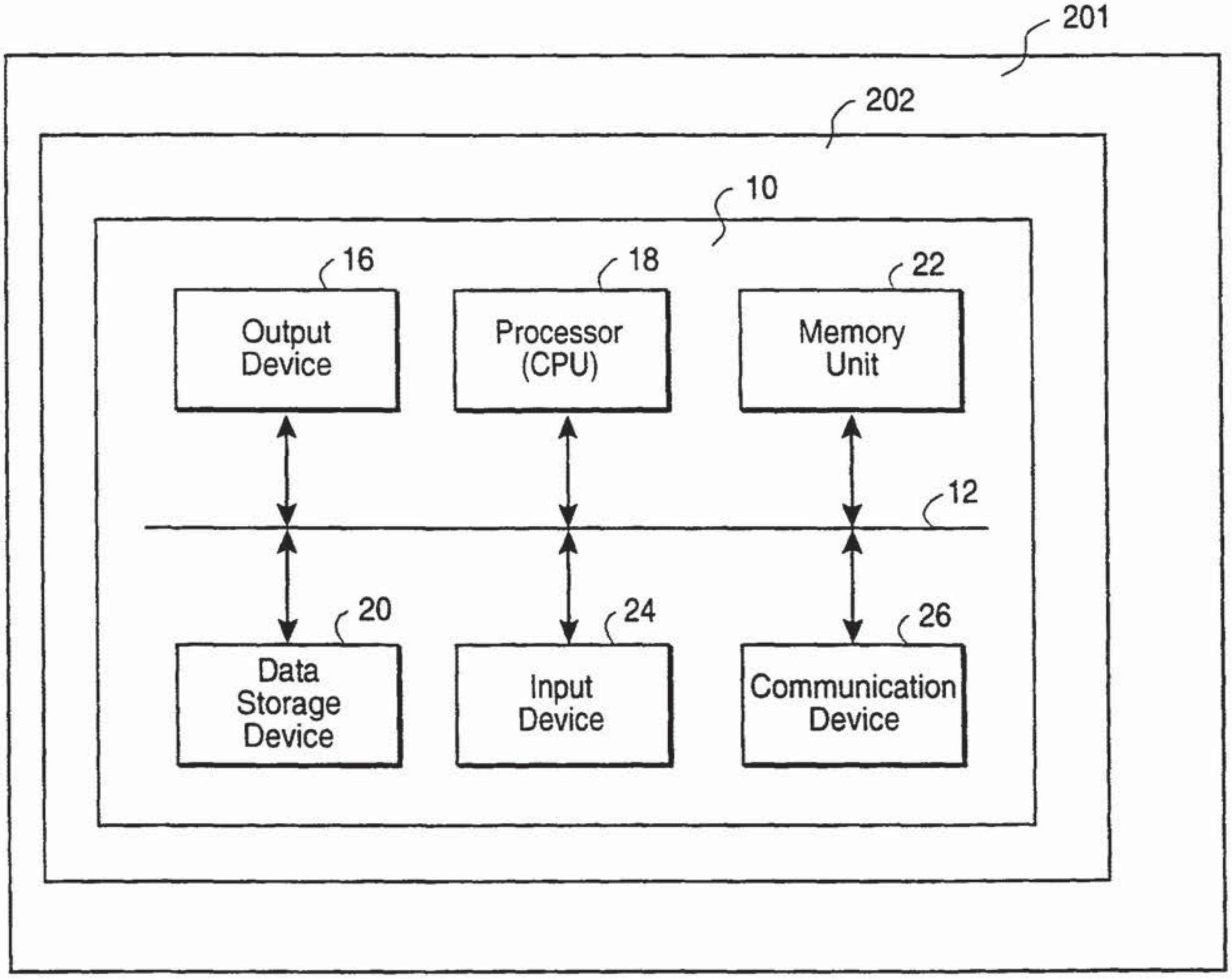
201
202
10
16
Output Device
18
Processor (CPU)
22
Memory Unit
12
20
Data Storage Device
24
Input Device
26
Communication Device

FIG. 1

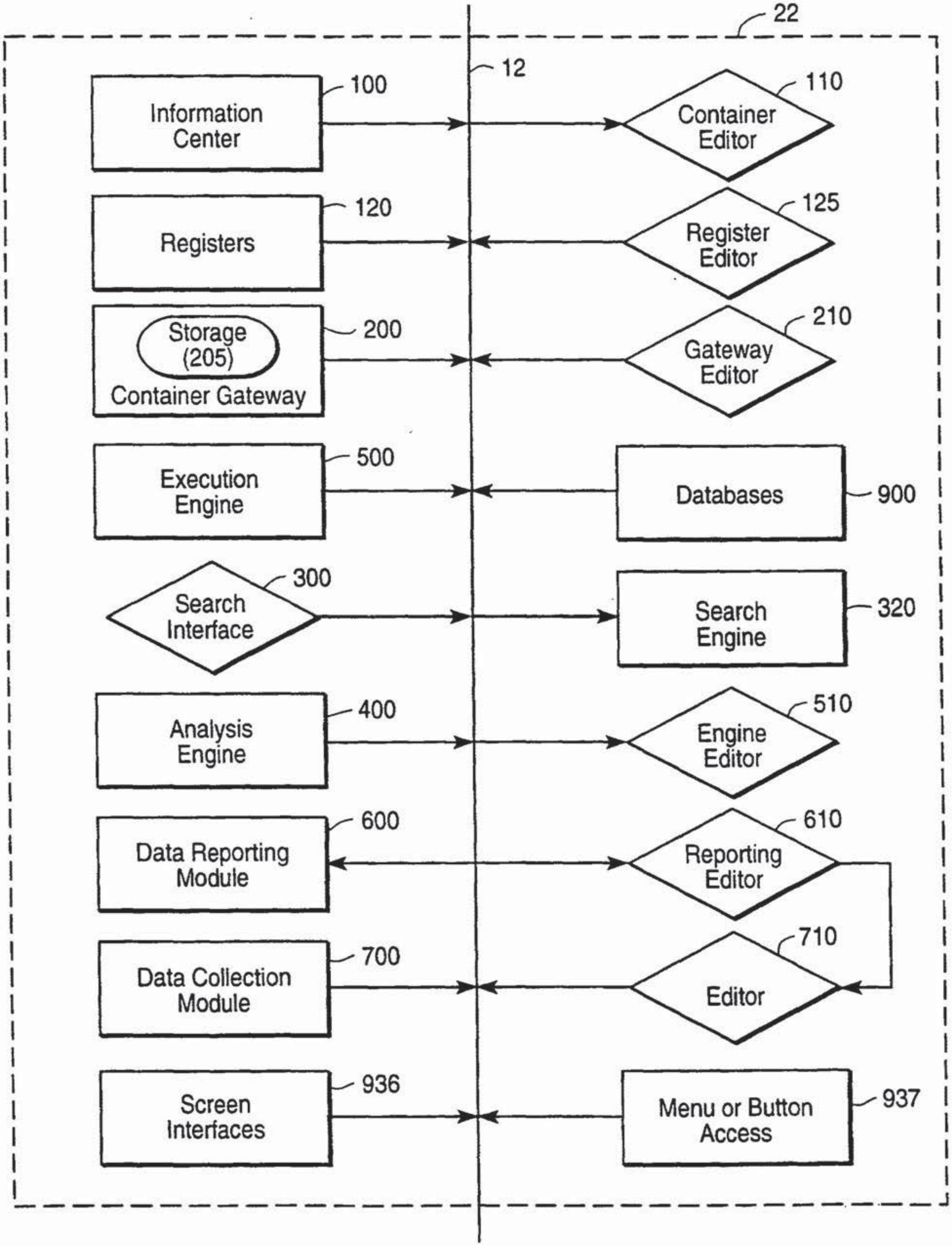
22
12
Information Center
100
Container Editor
110
Registers
120
Register Editor
125
Storage (205)
Container Gateway
200
Gateway Editor
210
Execution Engine
500
Databases
900
Search Interface
300
Search Engine
320
Analysis Engine
400
Engine Editor
510
Data Reporting Module
600
Reporting Editor
610
Data Collection Module
700
Editor
710
Screen Interfaces
936
Menu or Button Access
937

FIG. 2A

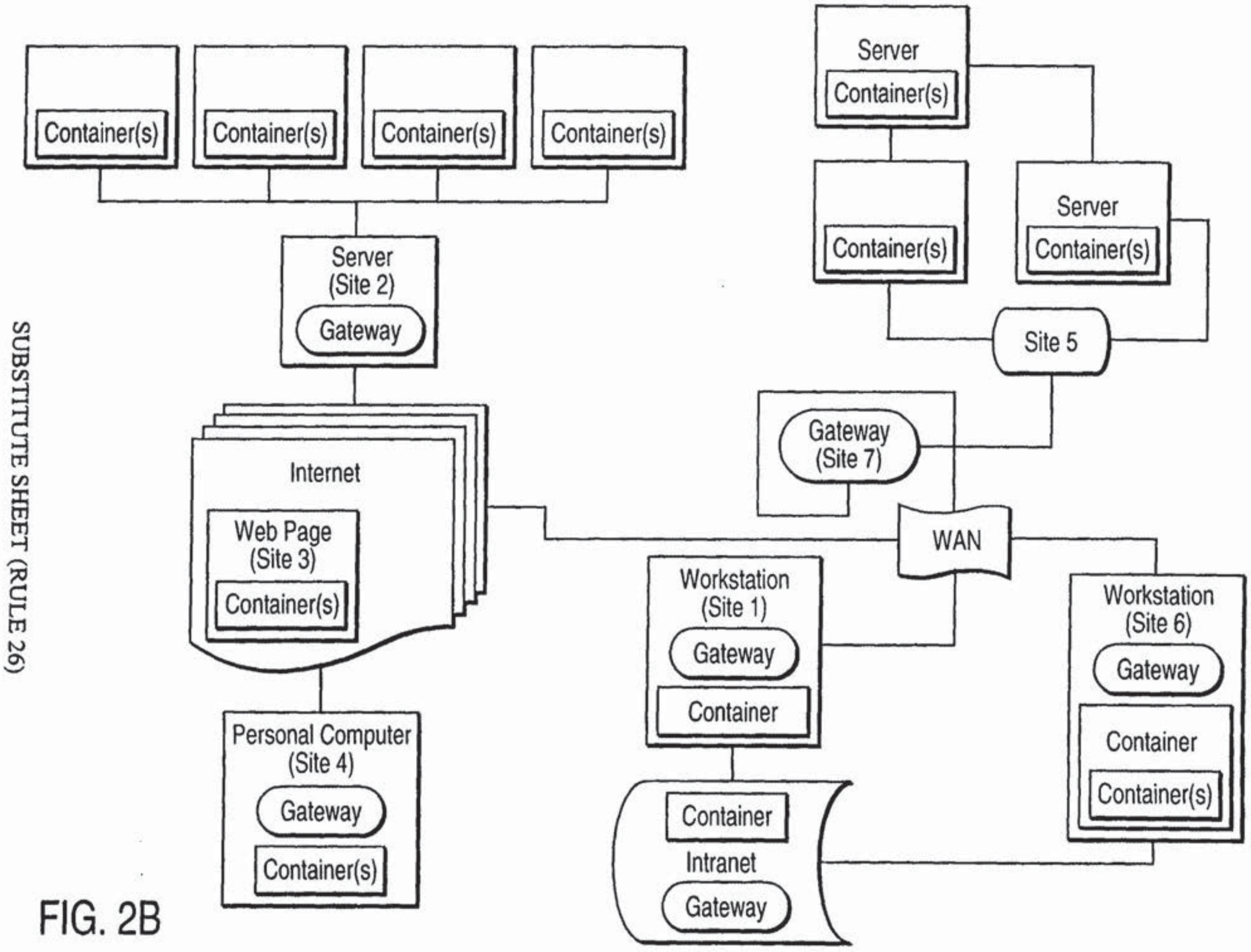
Container(s)
Container(s)
Container(s)
Container(s)
Server
(Site 2)
Gateway
Internet
Web Page
(Site 3)
Container(s)
Personal Computer
(Site 4)
Gateway
Container(s)
Server
Container(s)
Container(s)
Server
Container(s)
Site 5
Gateway
(Site 7)
WAN
Workstation
(Site 1)
Gateway
Container
Container
Intranet
Gateway
Workstation
(Site 6)
Gateway
Container
Container(s)

FIG. 2B

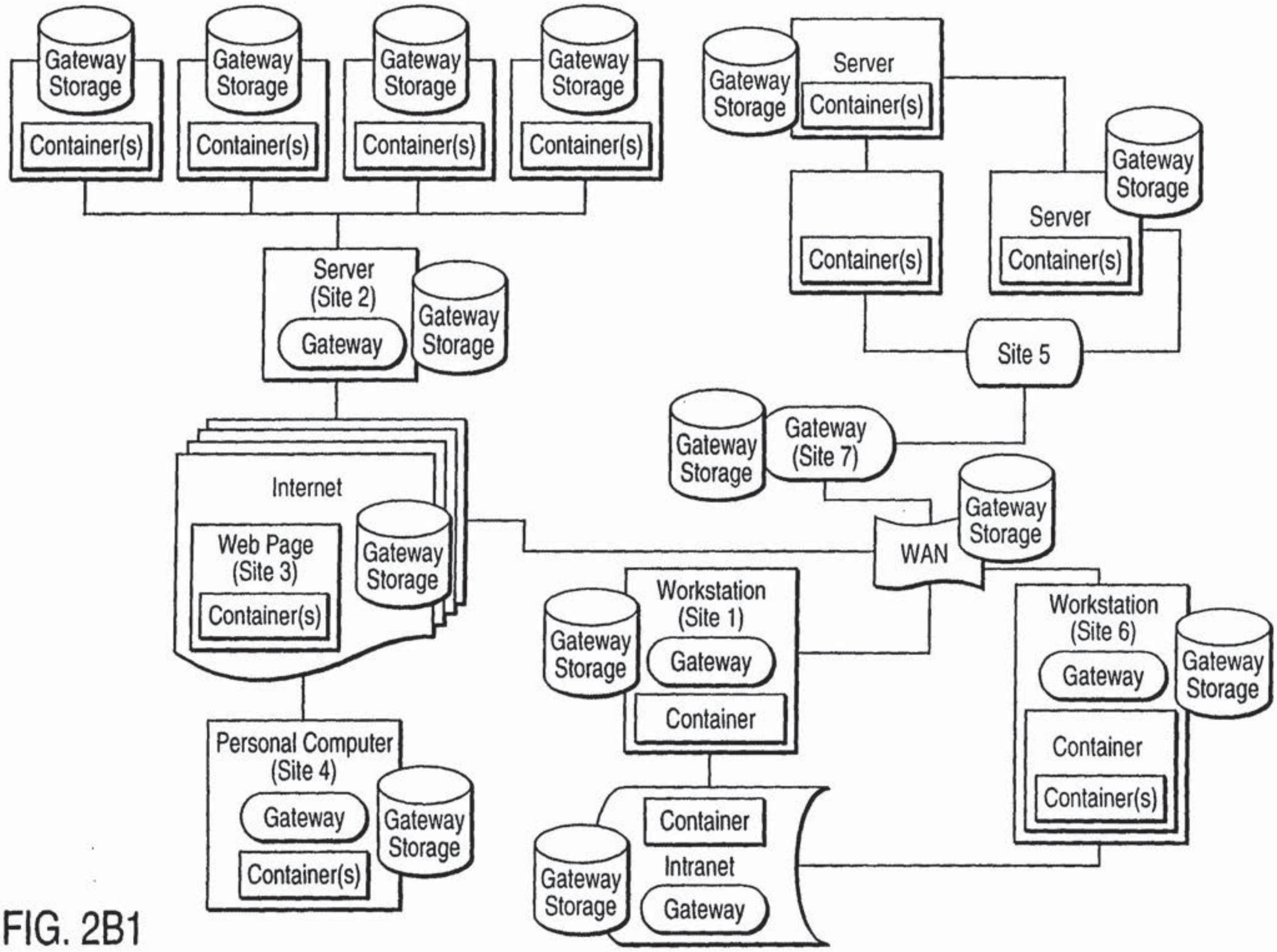
Gateway Storage
Container(s)
Gateway Storage
Container(s)
Gateway Storage
Container(s)
Gateway Storage
Container(s)
Gateway Storage
Server
Container(s)
Container(s)
Gateway Storage
Server
Container(s)
Server (Site 2)
Gateway
Gateway Storage
Site 5
Gateway Storage
Gateway (Site 7)
Internet
Web Page (Site 3)
Container(s)
Gateway Storage
WAN
Gateway Storage
Workstation (Site 1)
Gateway Storage
Gateway
Container
Workstation (Site 6)
Gateway
Gateway Storage
Container
Container(s)
Personal Computer (Site 4)
Gateway
Gateway Storage
Container(s)
Container
Intranet
Gateway Storage
Gateway

FIG. 2B1

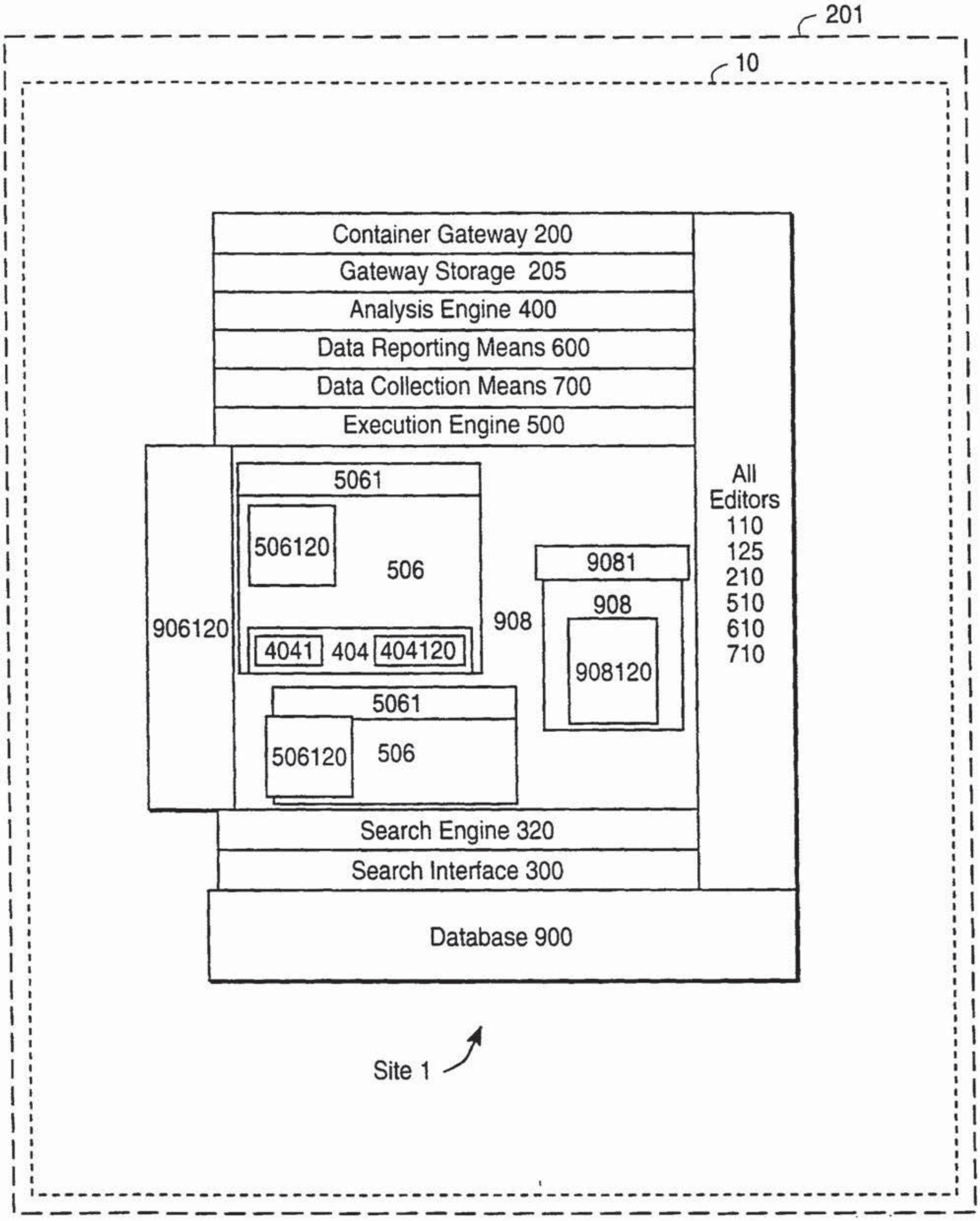
201
10
Container Gateway 200
Gateway Storage 205
Analysis Engine 400
Data Reporting Means 600
Data Collection Means 700
Execution Engine 500
906120
5061
506120
506
4041
404
404120
908
9081
908
908120
5061
506120
506
All
Editors
110
125
210
510
610
710
Search Engine 320
Search Interface 300
Database 900
Site 1

FIG. 2C

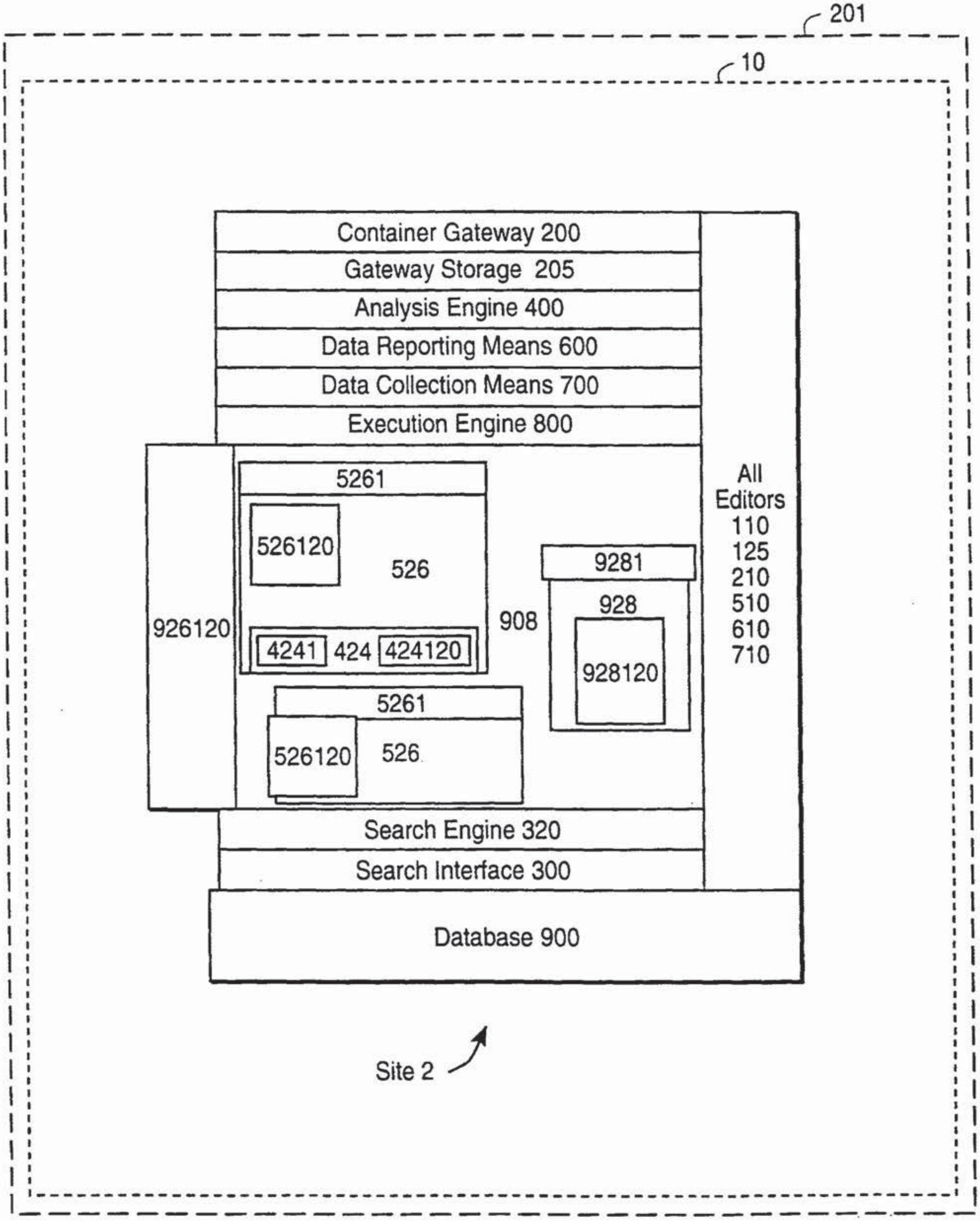
201
10
Container Gateway 200
Gateway Storage 205
Analysis Engine 400
Data Reporting Means 600
Data Collection Means 700
Execution Engine 800
5261
526120
526
926120
4241
424
424120
908
9281
928
928120
5261
526120
526
All Editors 110 125 210 510 610 710
Search Engine 320
Search Interface 300
Database 900
Site 2

FIG. 2D

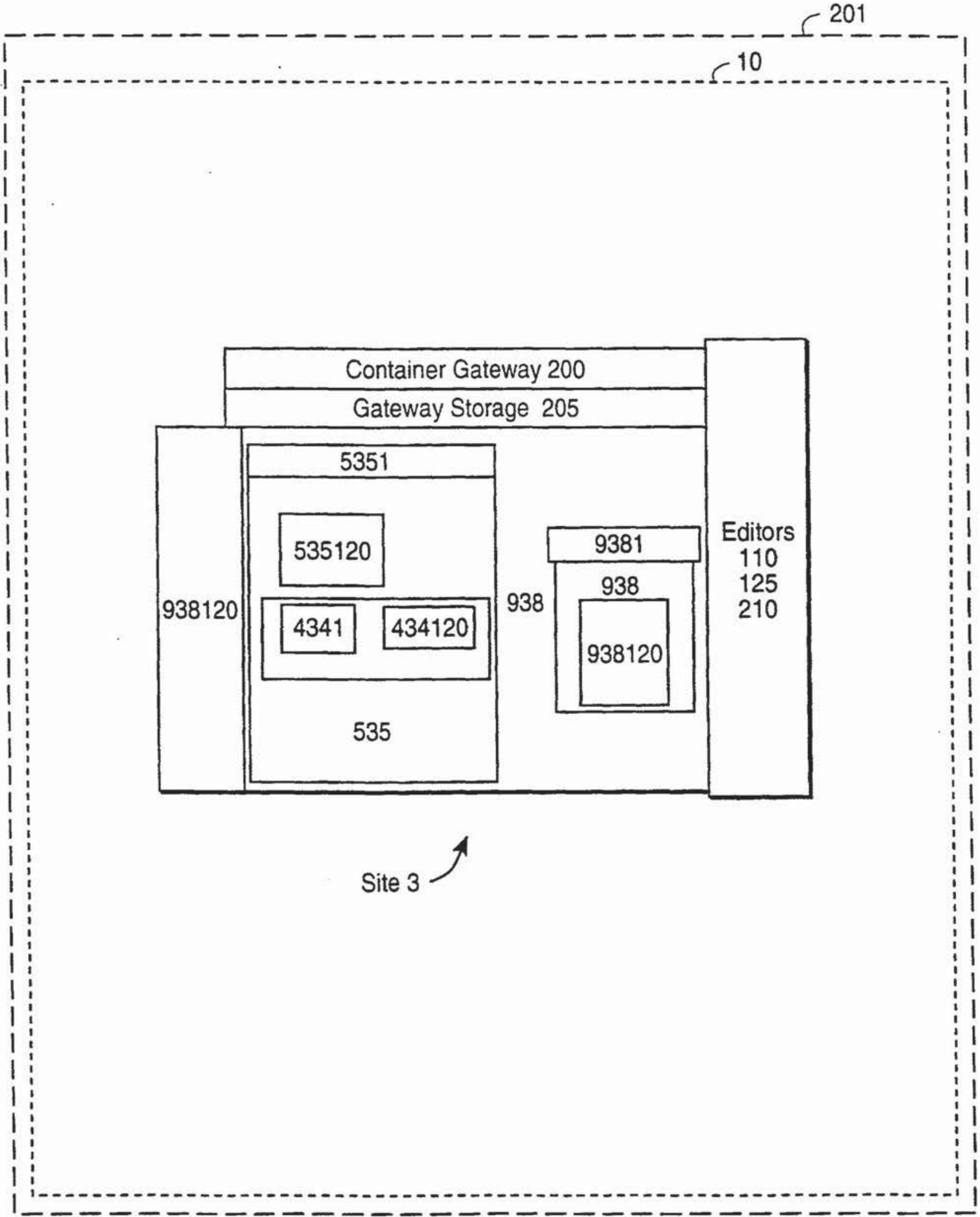
201
10
Container Gateway 200
Gateway Storage 205
5351
535120
938120
4341
434120
535
938
9381
938
938120
Editors
110
125
210
Site 3

FIG. 2E

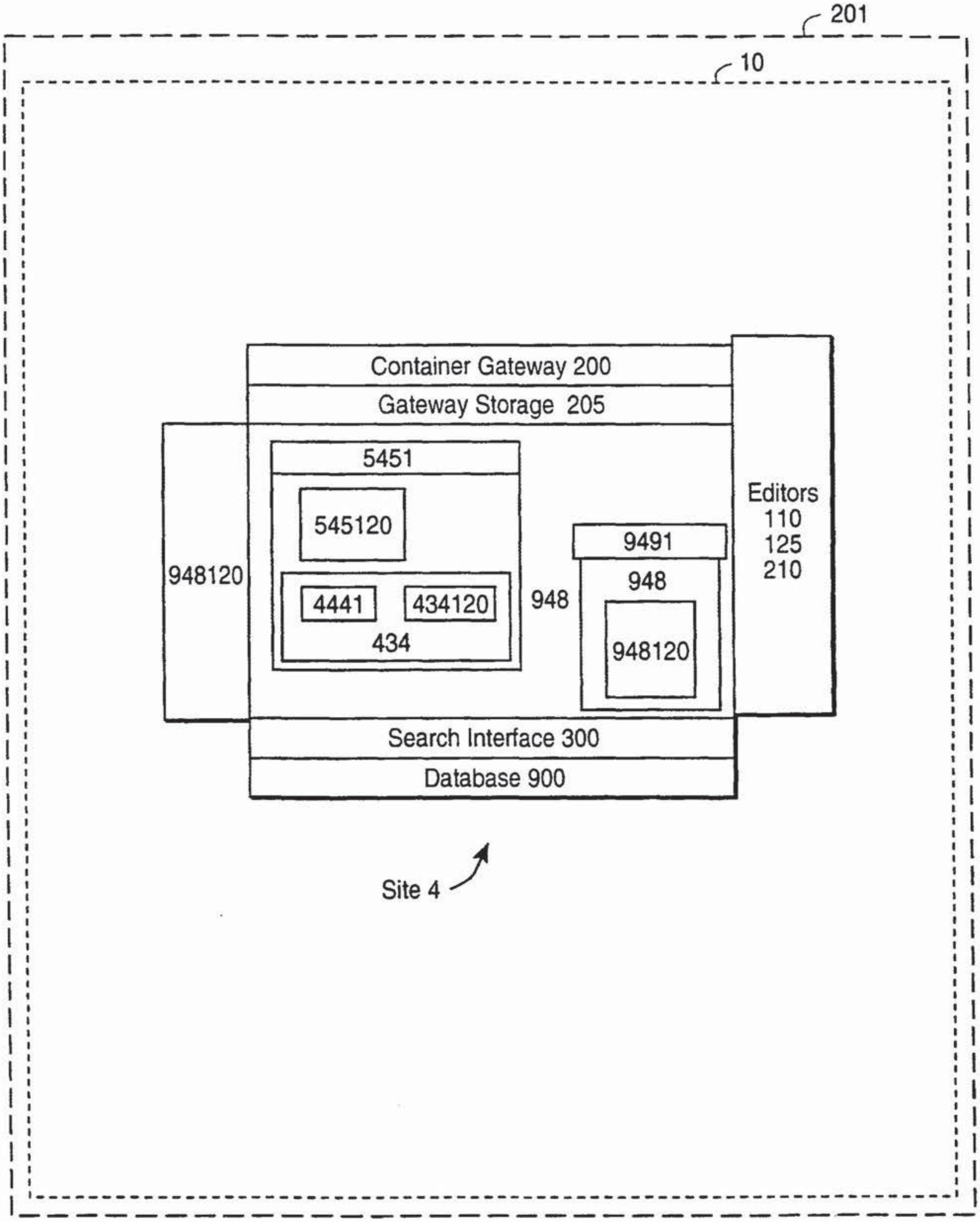
201
10
Container Gateway 200
Gateway Storage 205
5451
545120
948120
4441
434120
434
948
9491
948
948120
Editors
110
125
210
Search Interface 300
Database 900
Site 4

FIG. 2F

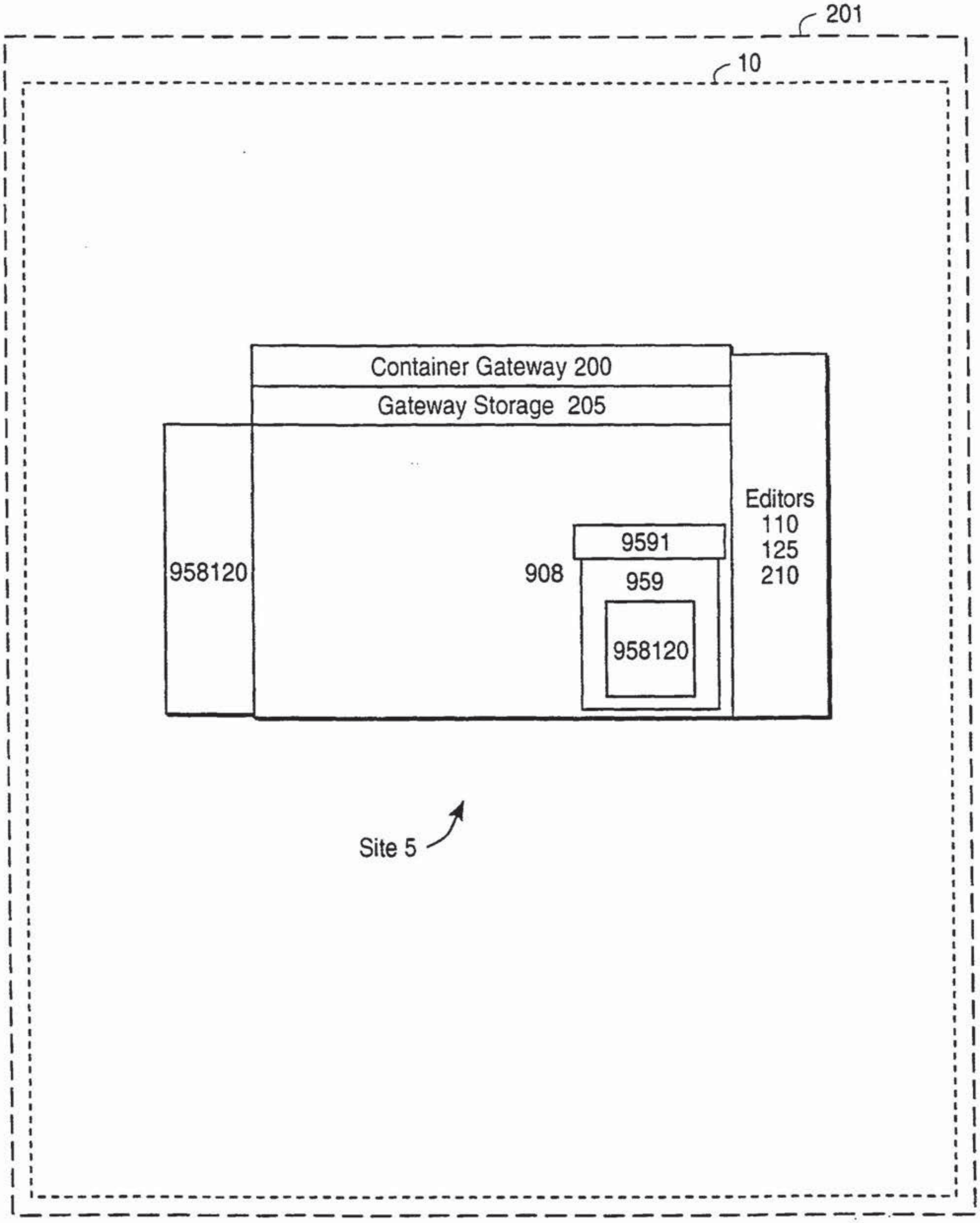
201
10
Container Gateway 200
Gateway Storage 205
958120
Editors
110
125
210
9591
908
959
958120
Site 5

FIG. 2G

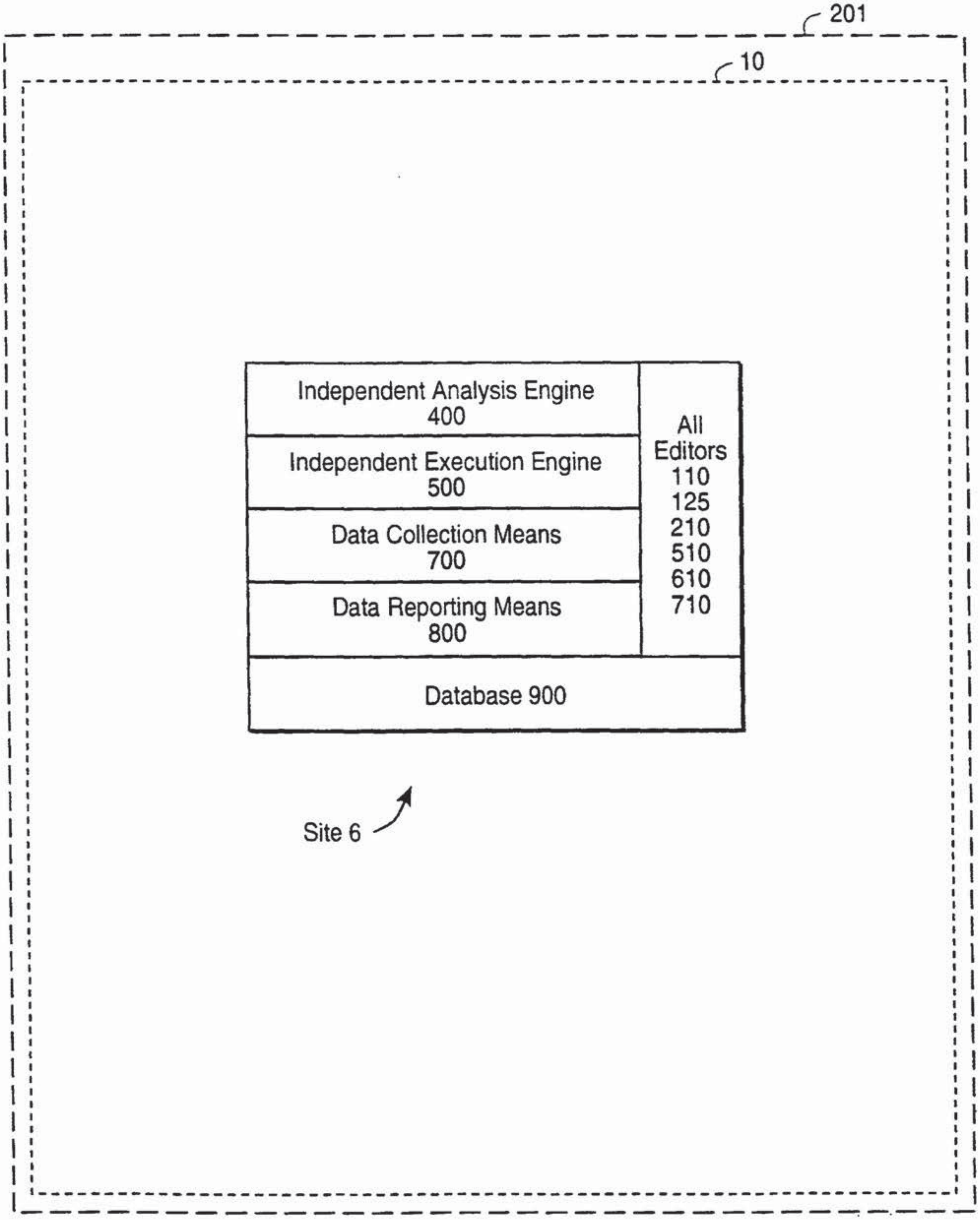
201
10
Independent Analysis Engine
400
Independent Execution Engine
500
Data Collection Means
700
Data Reporting Means
800
All
Editors
110
125
210
510
610
710
Database 900
Site 6

FIG. 2H

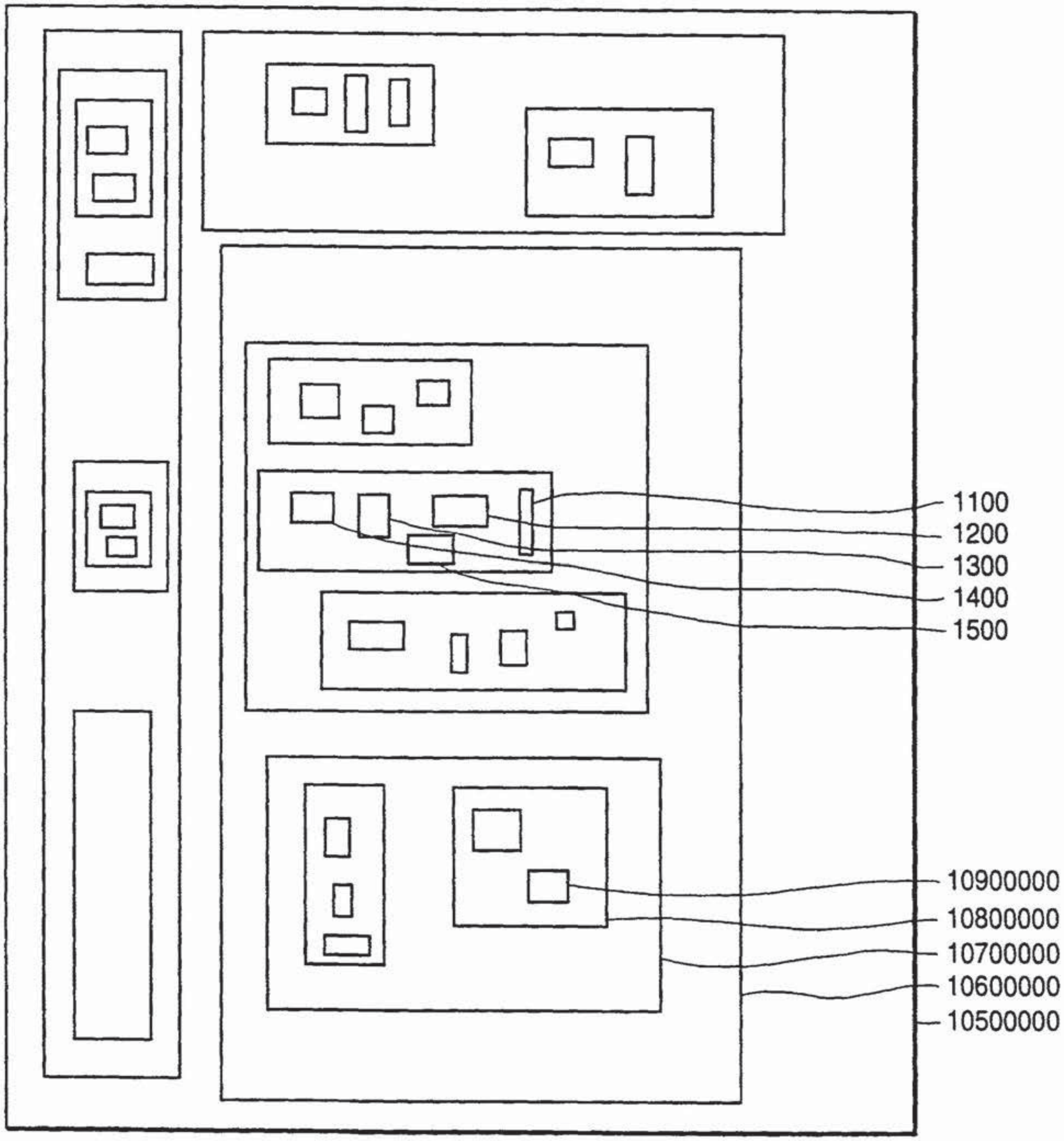
1100
1200
1300
1400
1500
10900000
10800000
10700000
10600000
10500000

FIG. 3A

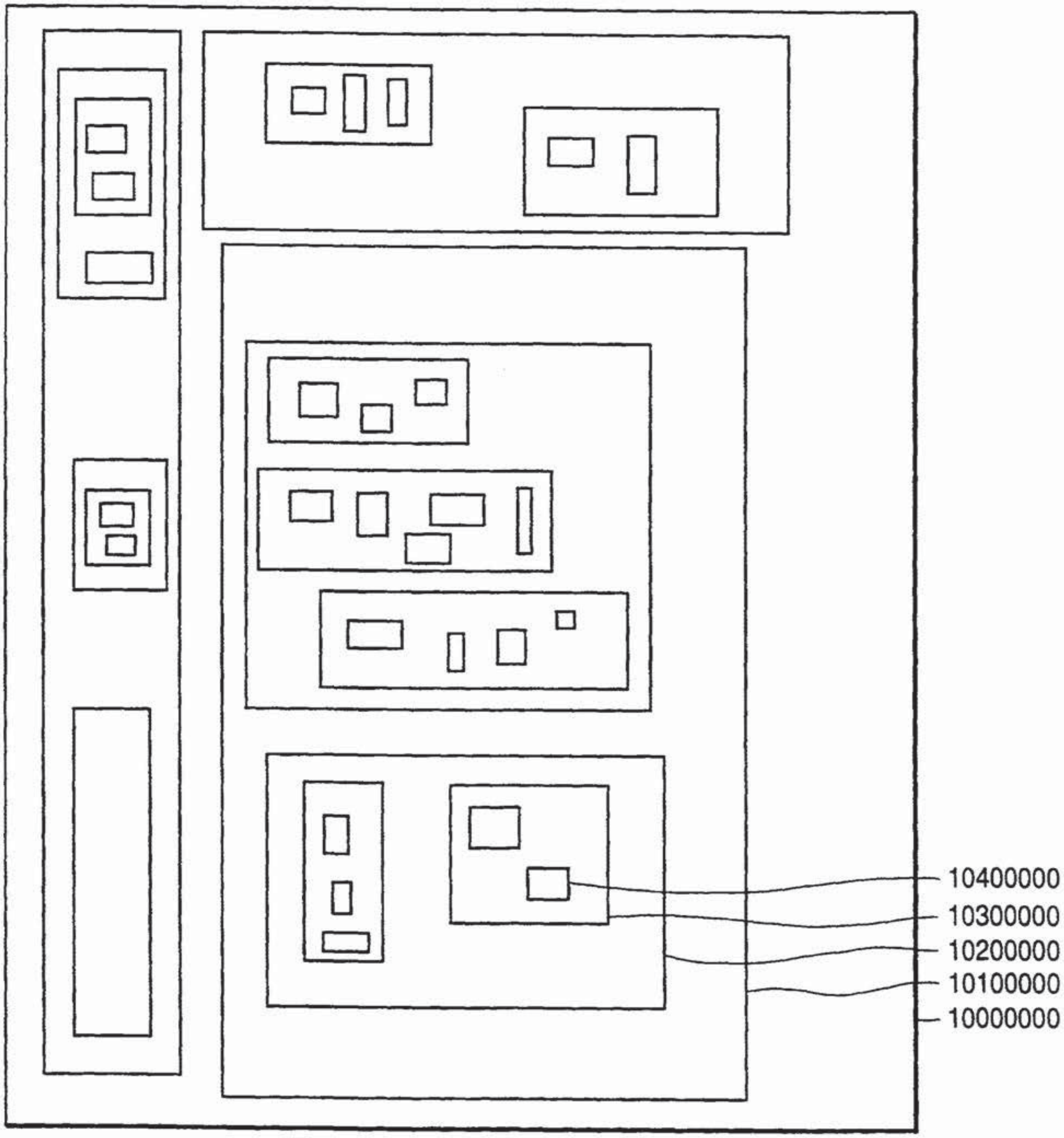

FIG. 3B

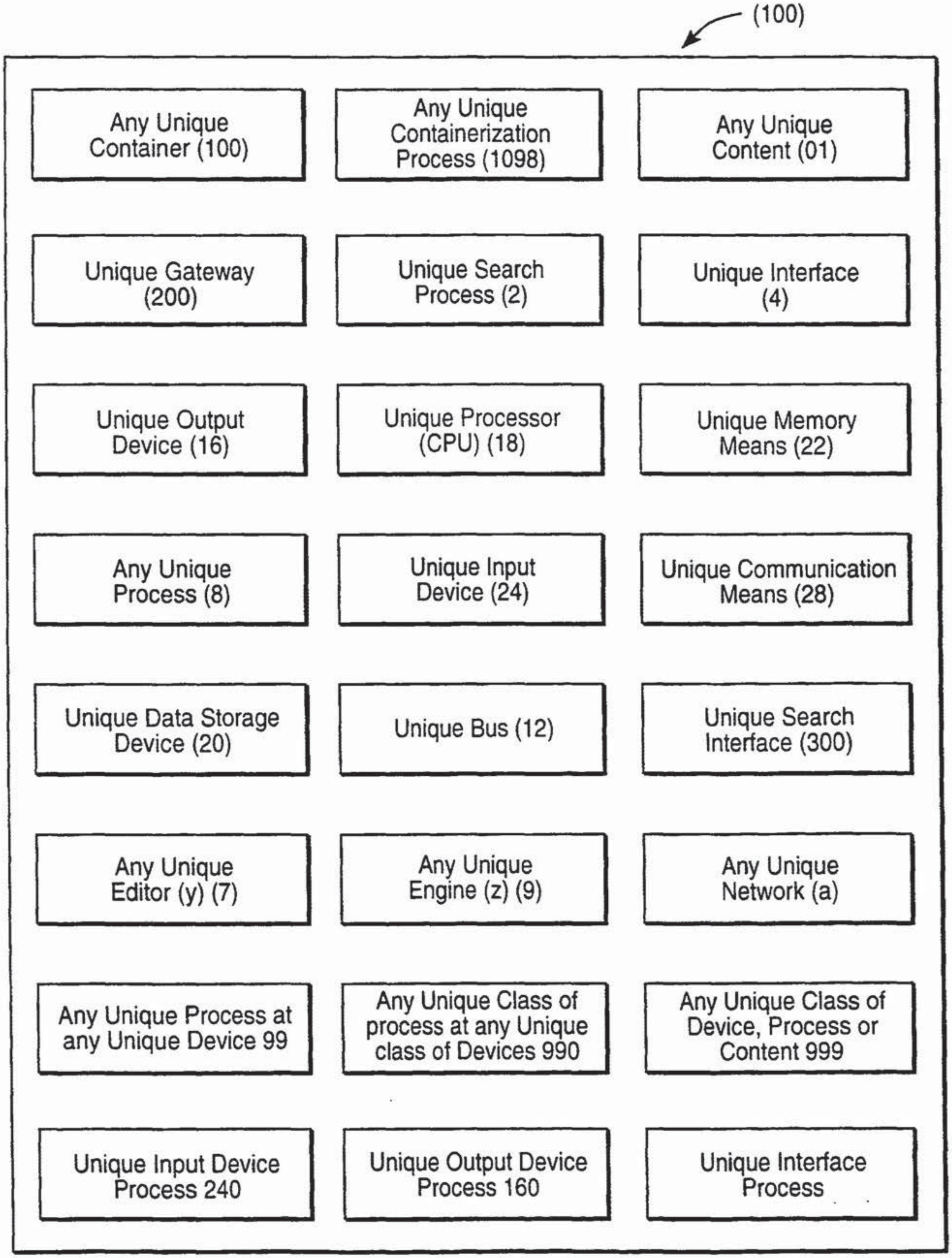
(100)
Any Unique Container (100)
Any Unique Containerization Process (1098)
Any Unique Content (01)
Unique Gateway (200)
Unique Search Process (2)
Unique Interface (4)
Unique Output Device (16)
Unique Processor (CPU) (18)
Unique Memory Means (22)
Any Unique Process (8)
Unique Input Device (24)
Unique 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

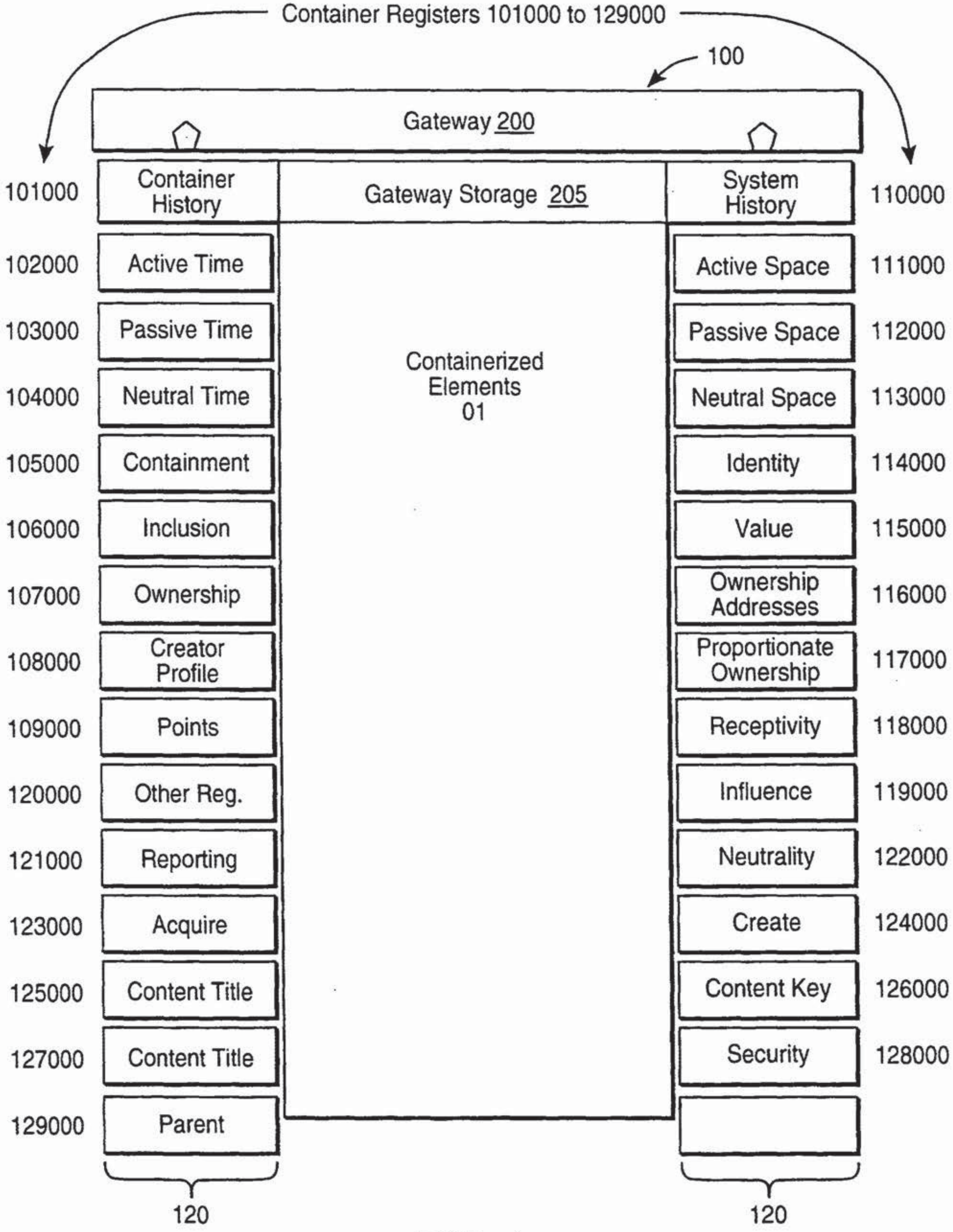
Container Registers 101000 to 129000
100
Gateway 200
101000
Container History
Gateway Storage 205
System History
110000
102000
Active Time
Active Space
111000
103000
Passive Time
Passive Space
112000
Containerized Elements 01
104000
Neutral Time
Neutral Space
113000
105000
Containment
Identity
114000
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 Title
Security
128000
129000
Parent
120
120

FIG. 4

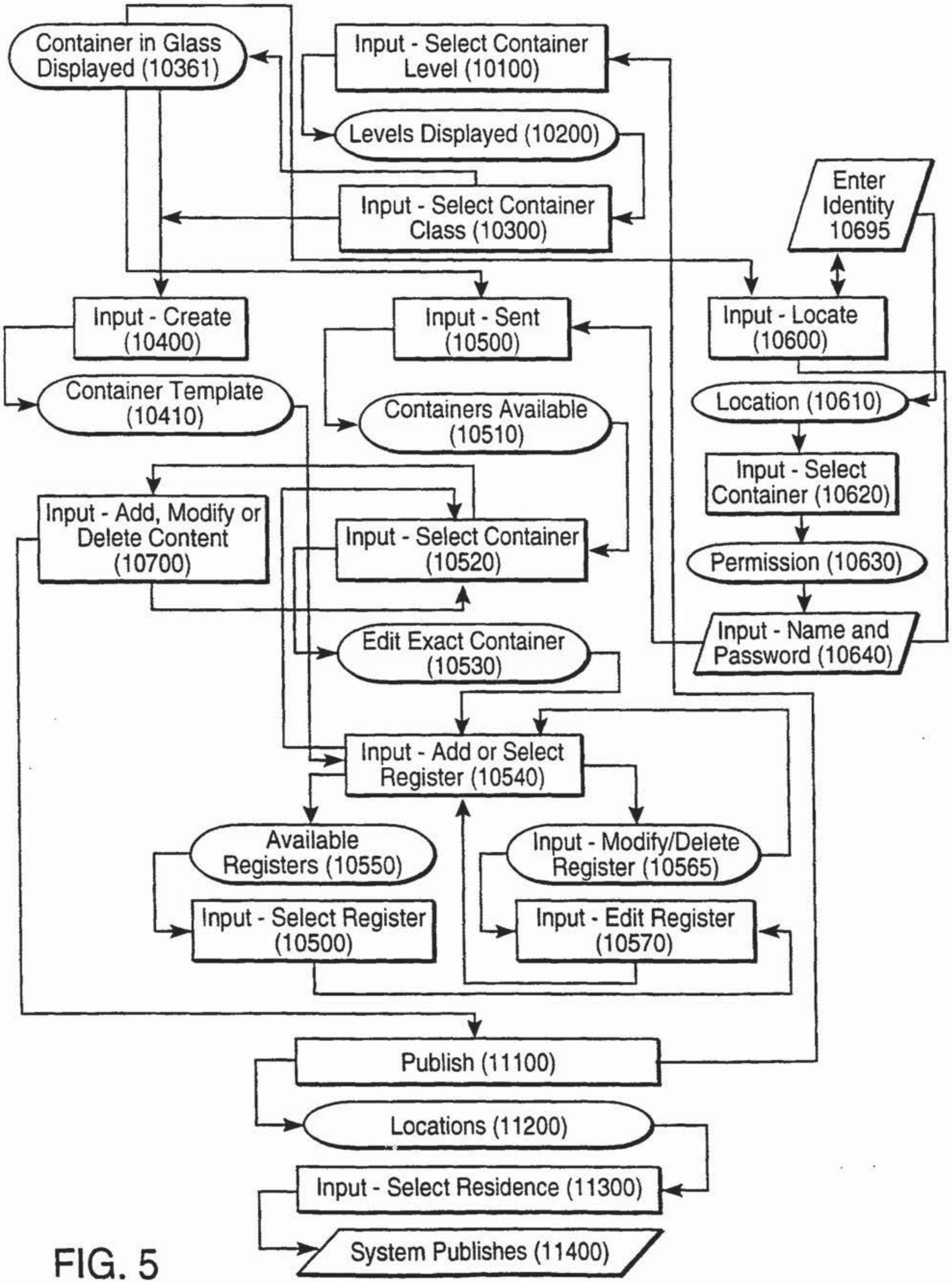
Container in Glass Displayed (10361)
Input - Select Container Level (10100)
Levels Displayed (10200)
Enter Identity 10695
Input - Select Container Class (10300)
Input - Create (10400)
Input - Sent (10500)
Input - Locate (10600)
Container Template (10410)
Containers Available (10510)
Location (10610)
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

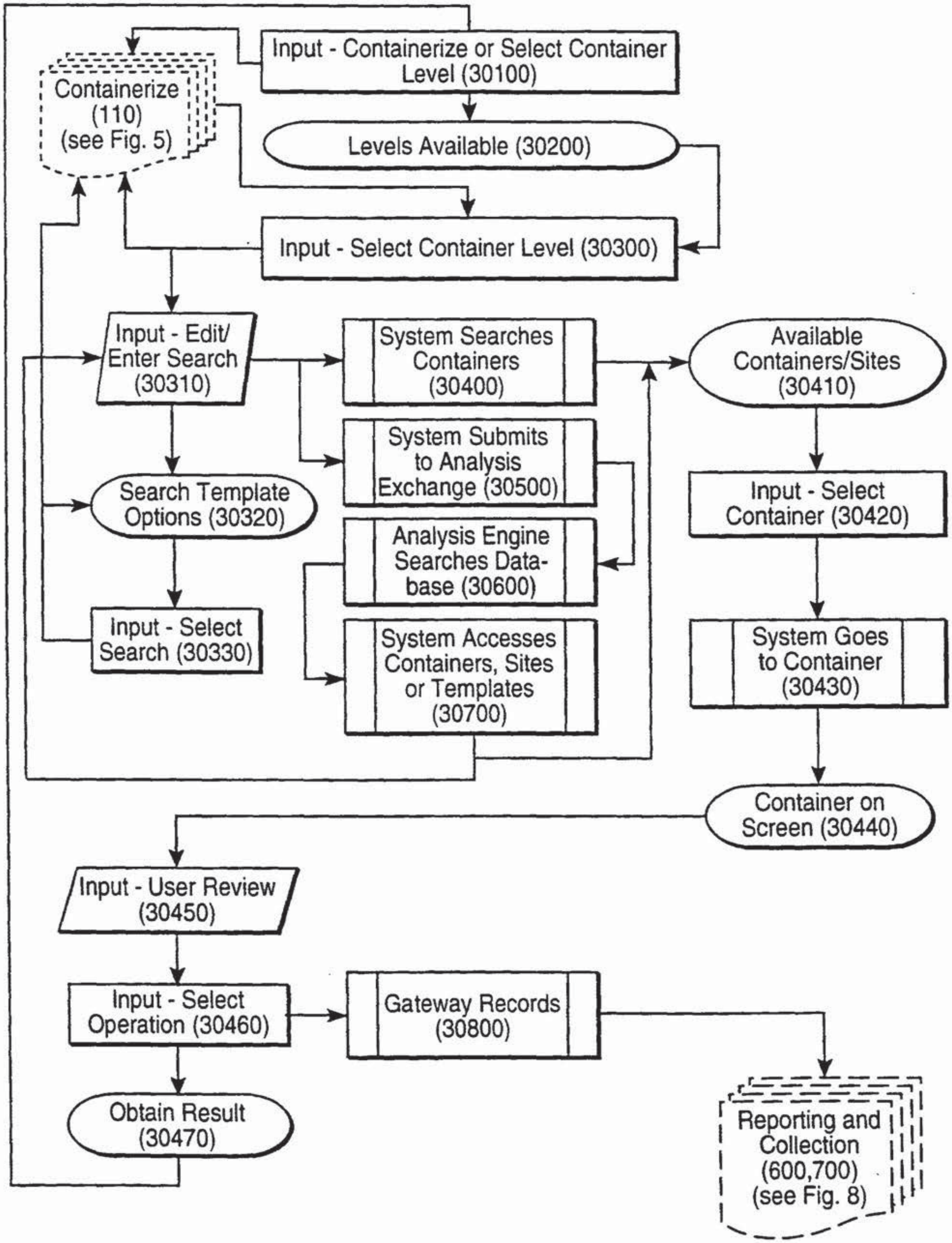
Input - Containerize or Select Container Level (30100)
Containerize (110) (see Fig. 5)
Levels Available (30200)
Input - Select Container Level (30300)
Input - Edit/ Enter Search (30310)
System Searches Containers (30400)
Available Containers/Sites (30410)
System Submits to Analysis Exchange (30500)
Search 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

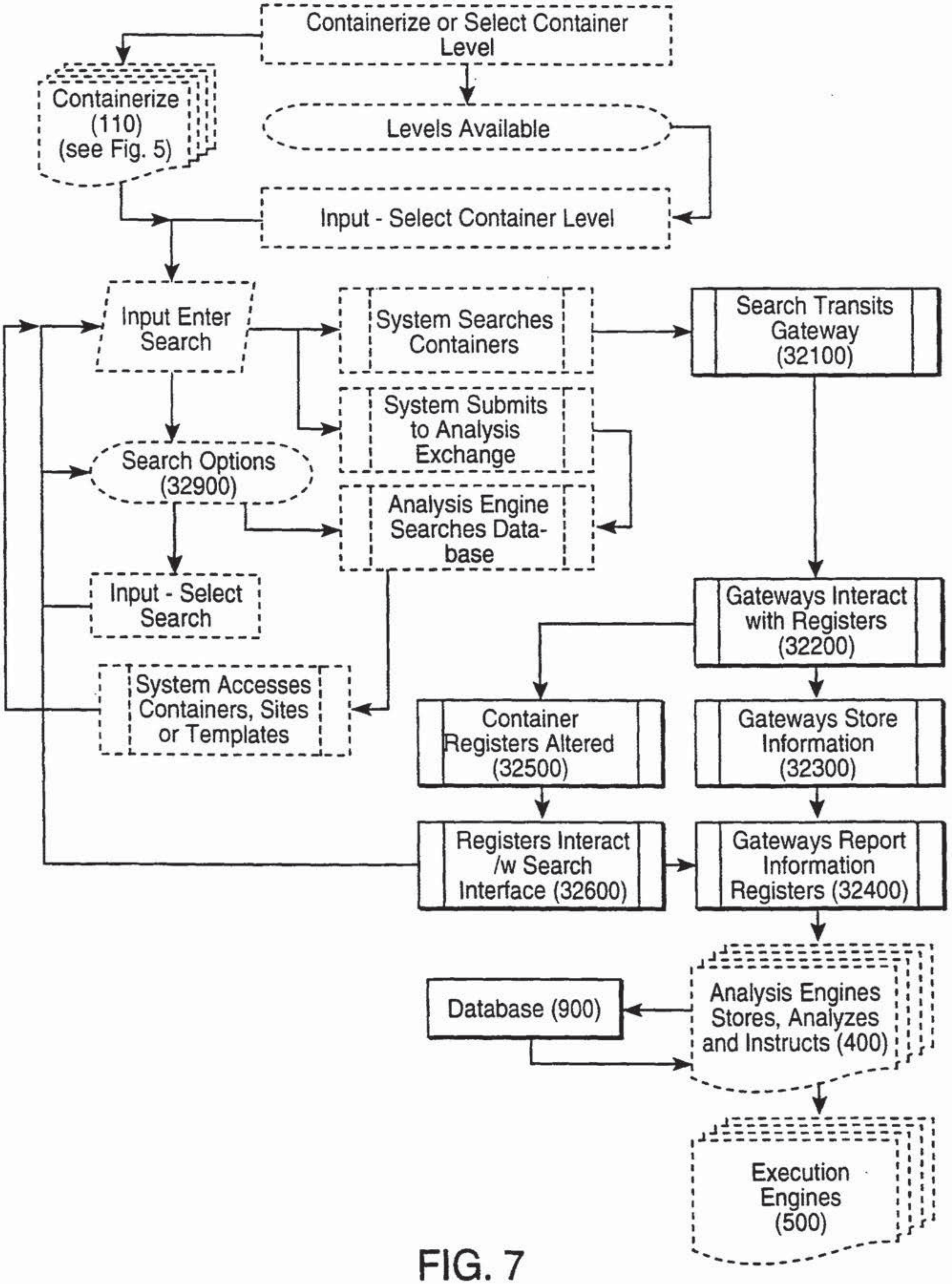
Containerize or Select Container Level
Containerize (110) (see Fig. 5)
Levels Available
Input - Select Container Level
Input Enter Search
System Searches Containers
Search Transits Gateway (32100)
System Submits to Analysis Exchange
Search Options (32900)
Analysis Engine Searches Data-base
Input - Select Search
System Accesses Containers, Sites or Templates
Gateways Interact with Registers (32200)
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

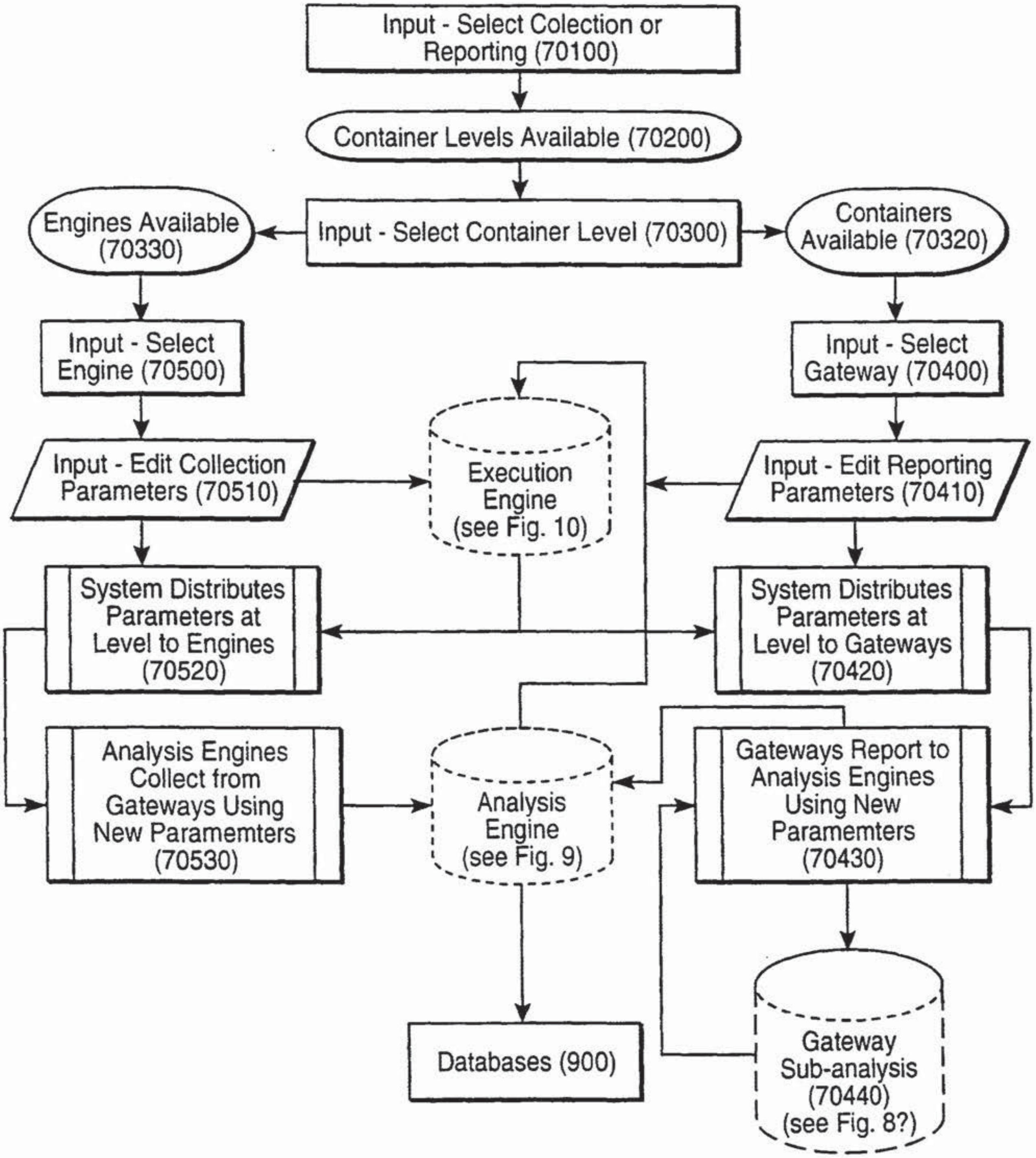
Input - Select Colection or Reporting (70100)
Container Levels Available (70200)
Input - Select Container Level (70300)
Engines Available (70330)
Containers Available (70320)
Input - Select Engine (70500)
Input - Select Gateway (70400)
Input - Edit Collection Parameters (70510)
Execution Engine (see 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

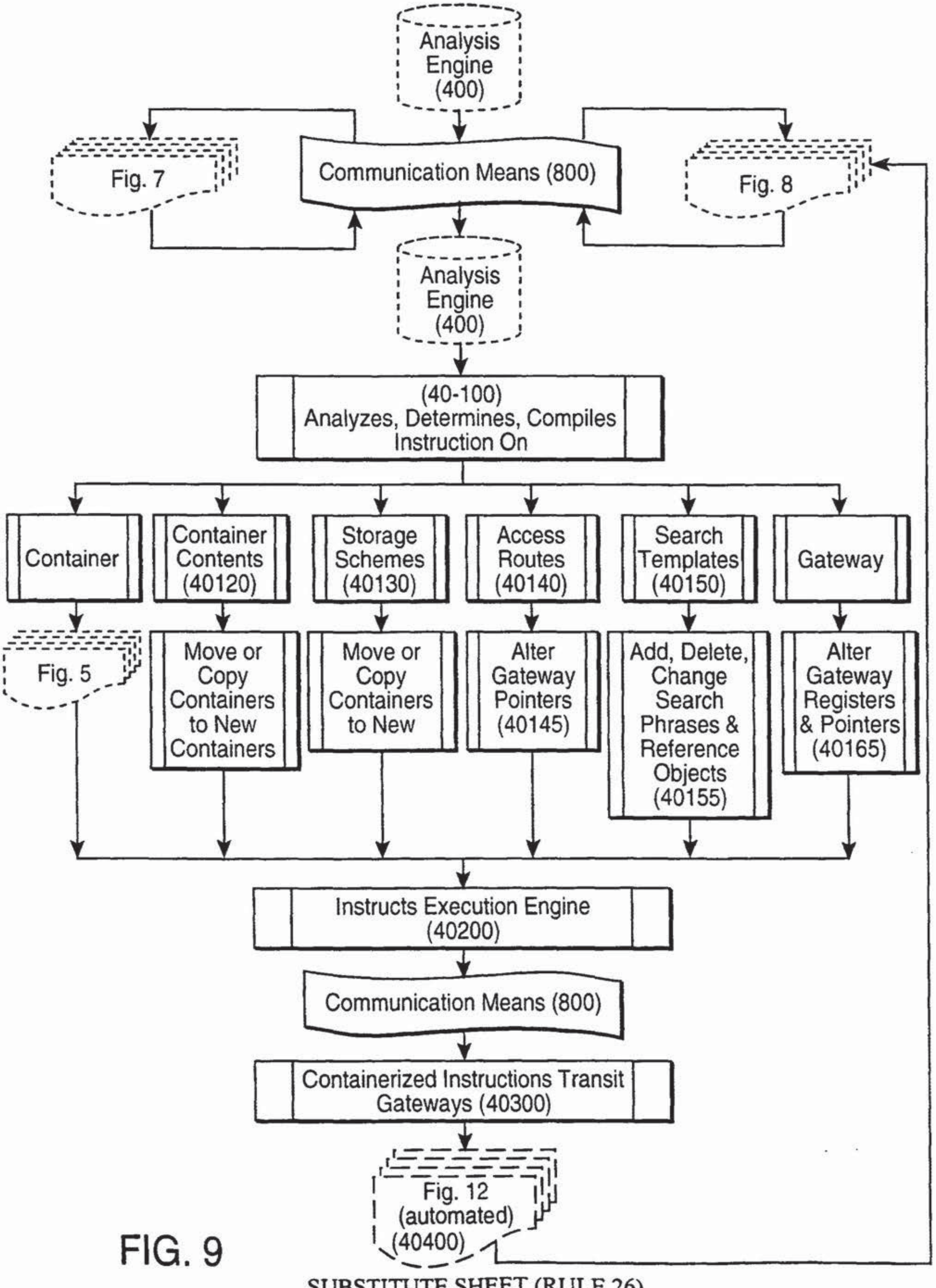
Analysis Engine (400)
Communication Means (800)
Fig. 7
Fig. 8
Analysis Engine (400)
(40-100) Analyzes, Determines, Compiles Instruction On
Container
Container Contents (40120)
Storage Schemes (40130)
Access Routes (40140)
Search Templates (40150)
Gateway
Fig. 5
Move or Copy Containers to New Containers
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

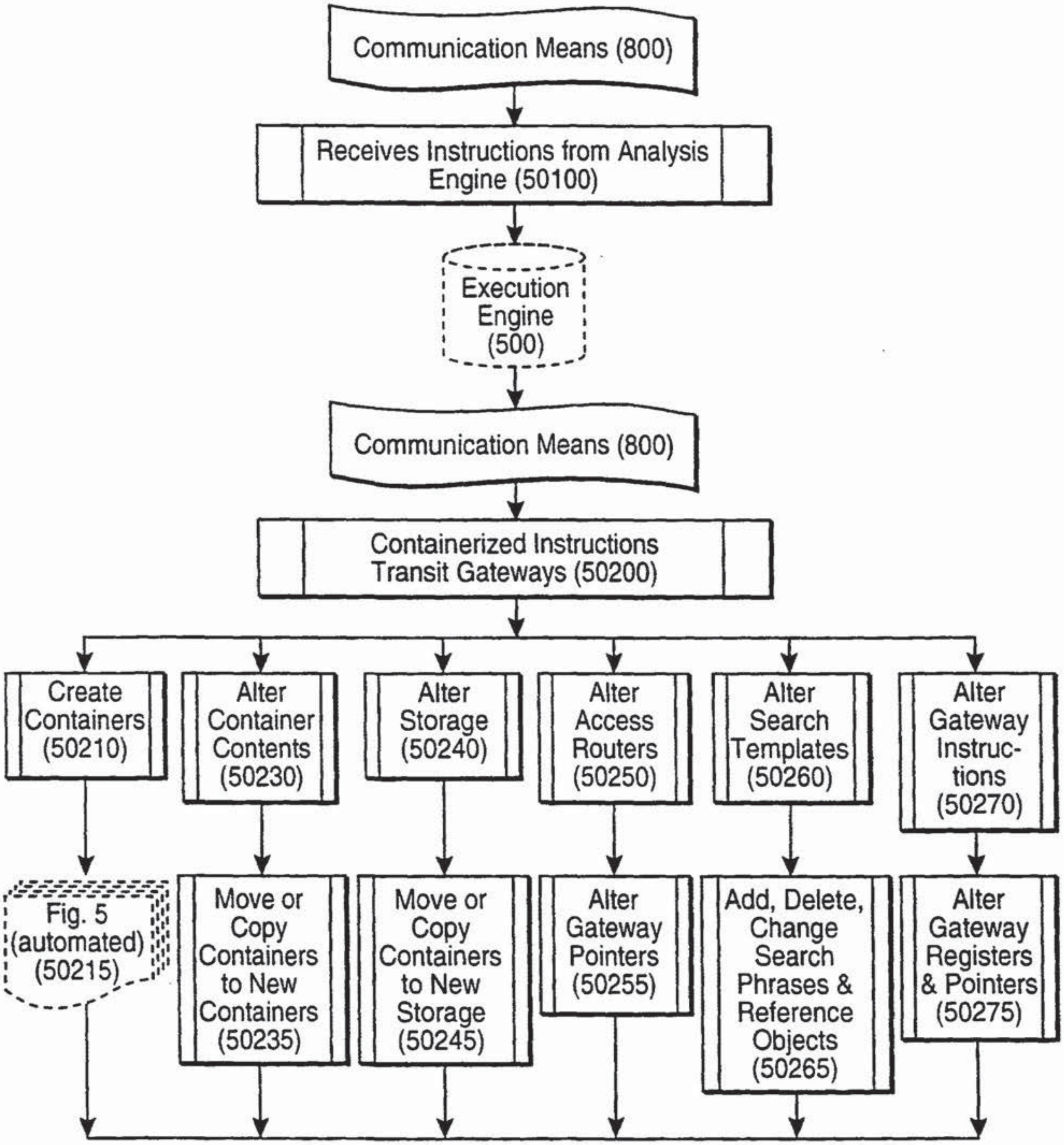
Communication Means (800)
Receives Instructions from Analysis Engine (50100)
Execution Engine (500)
Communication Means (800)
Containerized Instructions Transit Gateways (50200)
Create Containers (50210)
Alter Container Contents (50230)
Alter Storage (50240)
Alter Access Routers (50250)
Alter Search 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

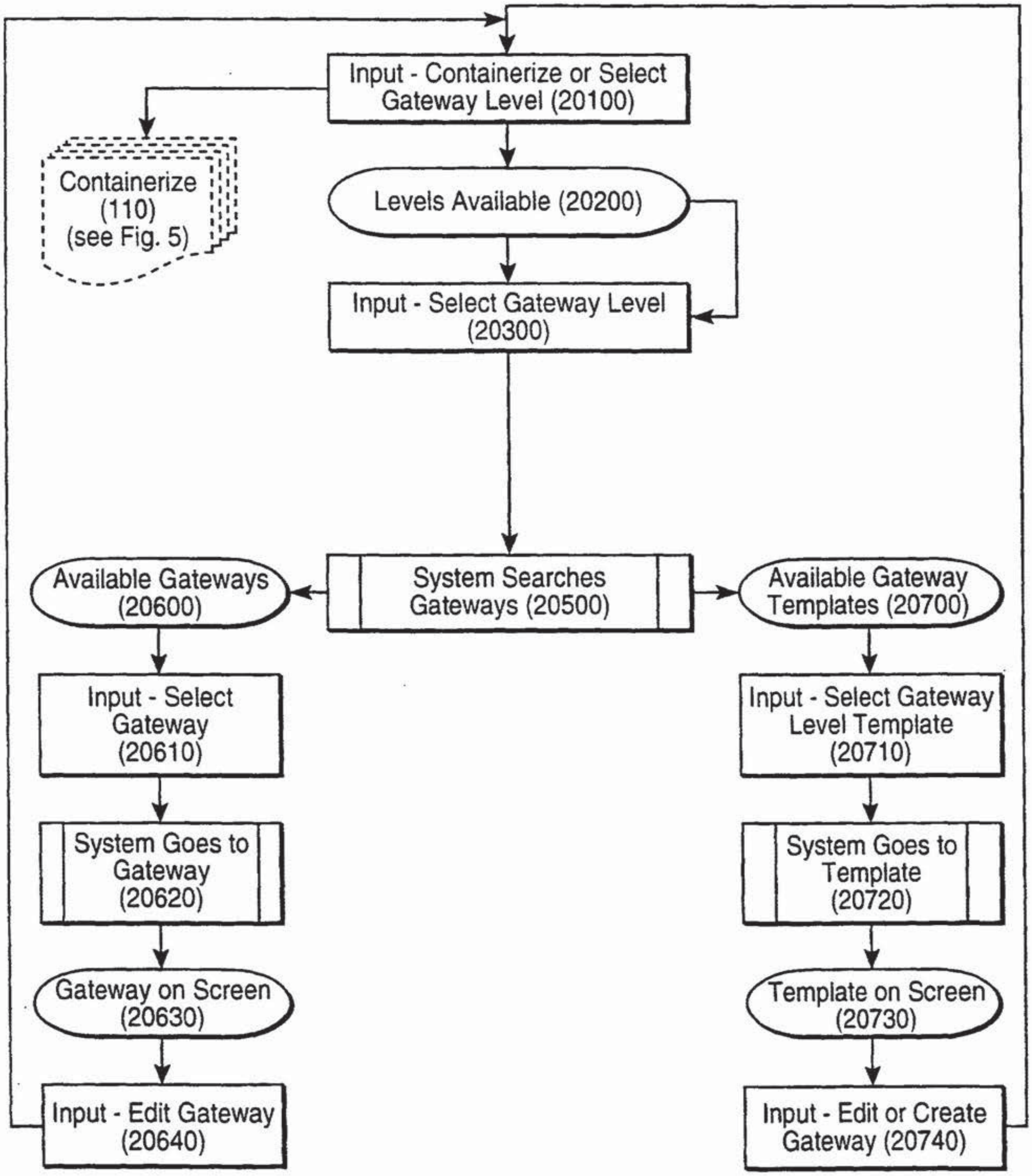
Input - Containerize or Select Gateway Level (20100)
Containerize (110) (see Fig. 5)
Levels Available (20200)
Input - Select Gateway Level (20300)
System Searches Gateways (20500)
Available Gateways (20600)
Input - Select Gateway (20610)
System Goes to Gateway (20620)
Gateway on Screen (20630)
Input - Edit Gateway (20640)
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

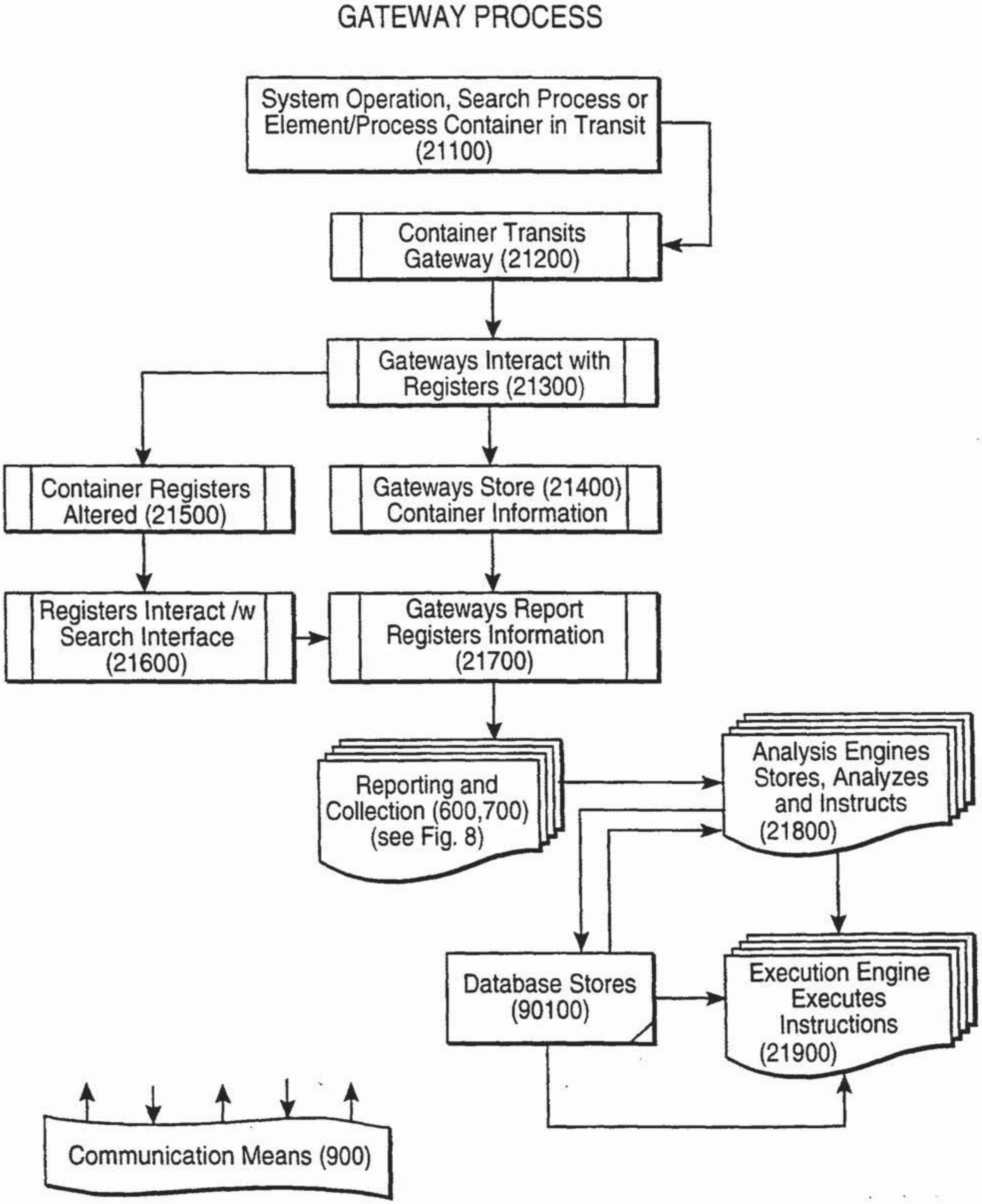
GATEWAY PROCESS
System Operation, Search Process or Element/Process Container in Transit (21100)
Container Transits Gateway (21200)
Gateways Interact with Registers (21300)
Container Registers Altered (21500)
Gateways Store (21400) Container Information
Registers Interact /w Search Interface (21600)
Gateways Report Registers Information (21700)
Reporting and Collection (600,700) (see Fig. 8)
Analysis Engines Stores, Analyzes and Instructs (21800)
Database Stores (90100)
Execution Engine Executes Instructions (21900)
Communication Means (900)

FIG. 12

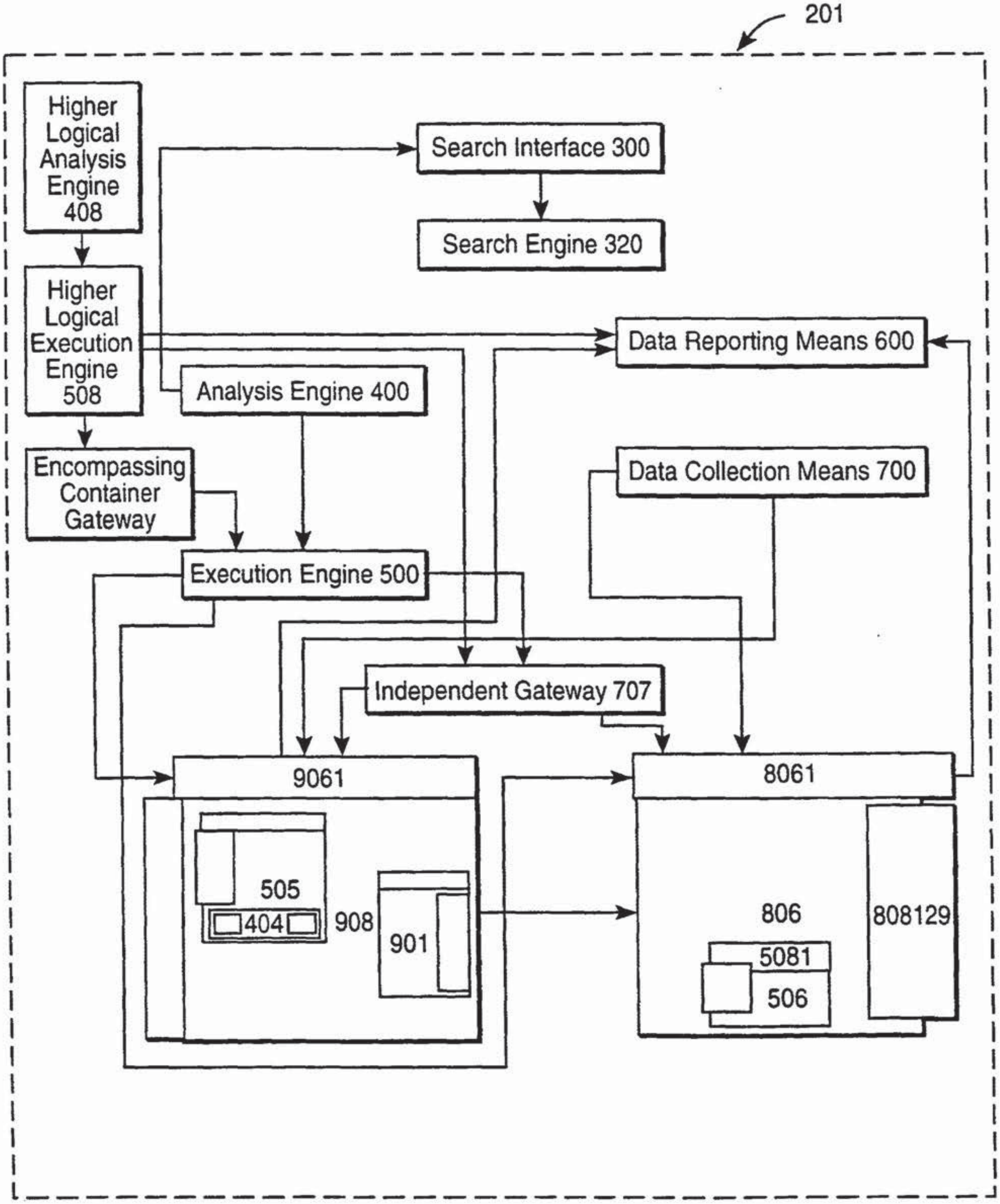
201
Higher Logical Analysis Engine 408
Higher Logical Execution Engine 508
Encompassing Container Gateway
Search Interface 300
Search Engine 320
Data Reporting Means 600
Analysis Engine 400
Data Collection Means 700
Execution Engine 500
Independent Gateway 707
9061
8061
505
404
908
901
806
808129
5081
506

FIG. 13A

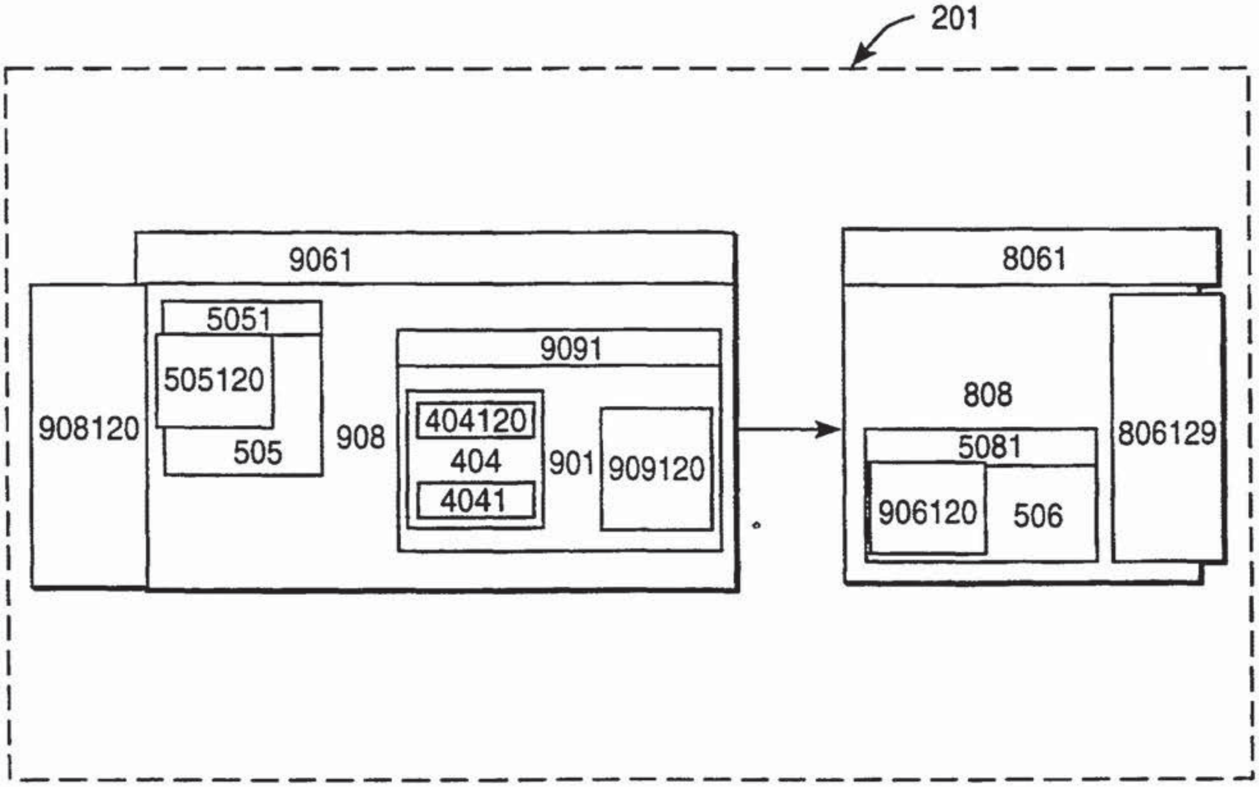
201
9061
8061
5051
9091
505120
808
404120
908120
908
404
901
909120
505
5081
806129
4041
906120
506

FIG. 13B

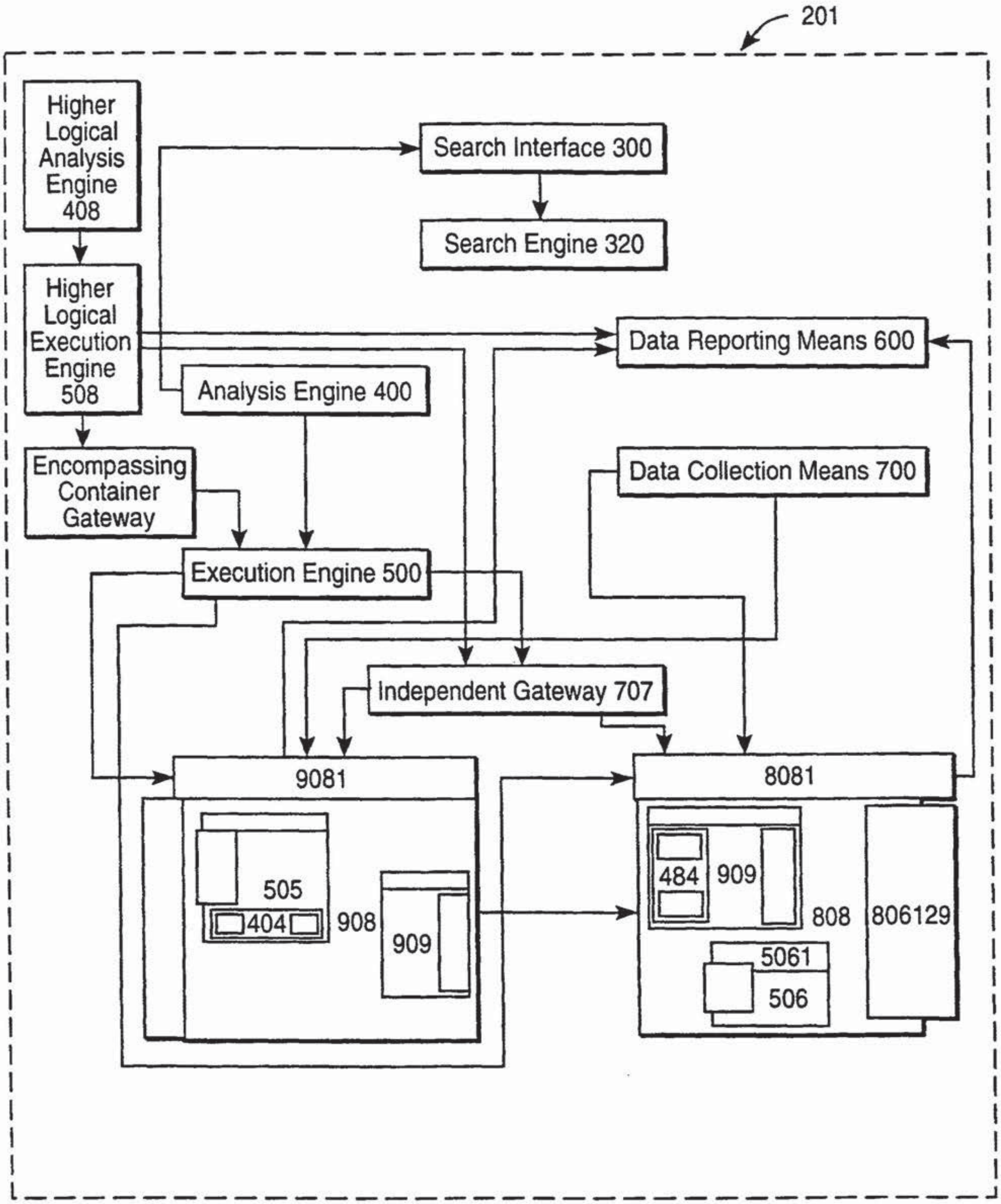
201
Higher Logical Analysis Engine 408
Search Interface 300
Search Engine 320
Higher Logical Execution Engine 508
Data Reporting Means 600
Analysis Engine 400
Data Collection Means 700
Encompassing Container Gateway
Execution Engine 500
Independent Gateway 707
9081
8081
505
404
908
909
484
909
808
806129
5061
506

FIG. 13C

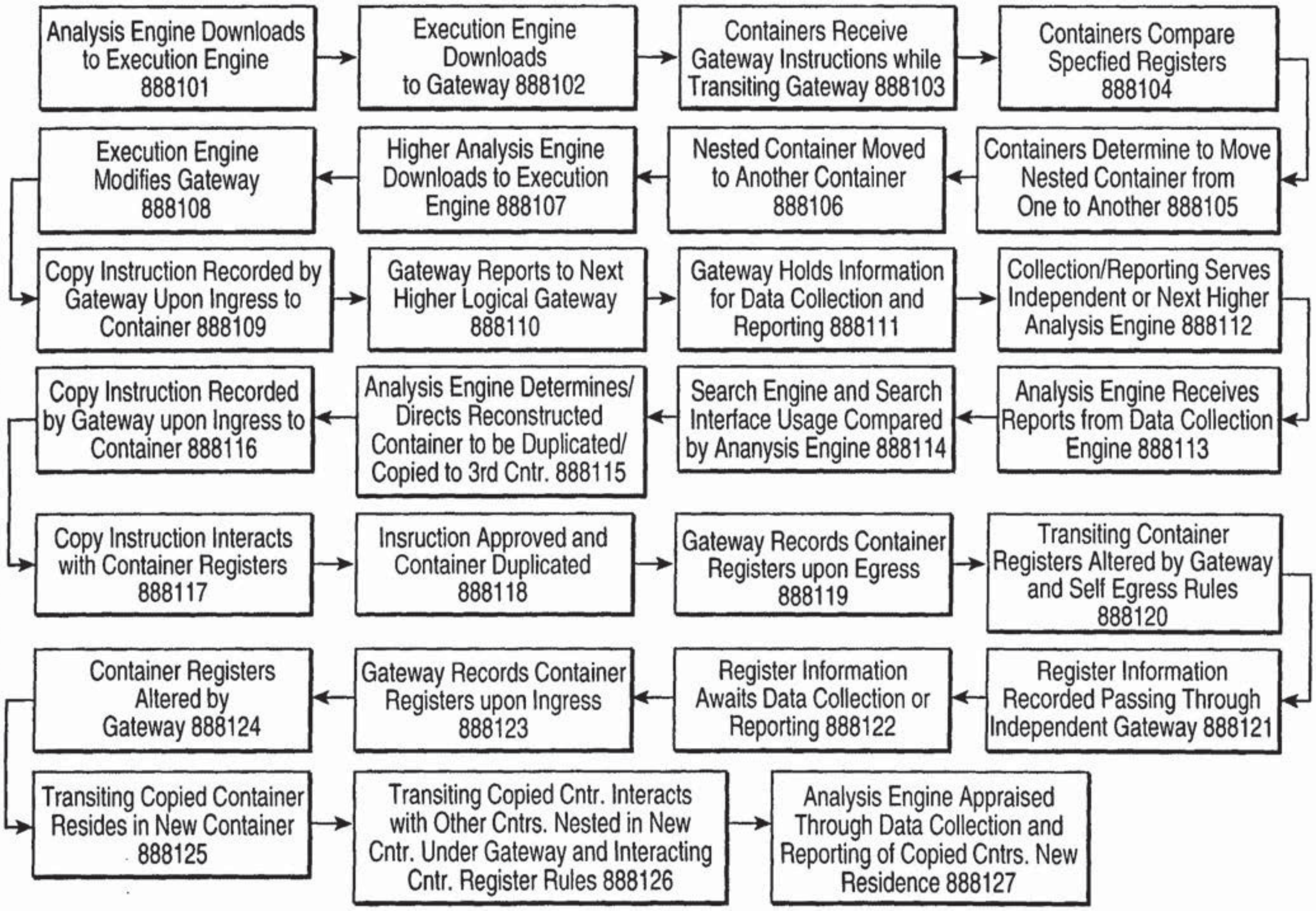
Analysis Engine Downloads to Execution Engine 888101
Execution Engine Downloads to Gateway 888102
Containers Receive Gateway Instructions while Transiting Gateway 888103
Containers Compare Specfied Registers 888104
Containers Determine to Move Nested Container from One to Another 888105
Nested Container 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

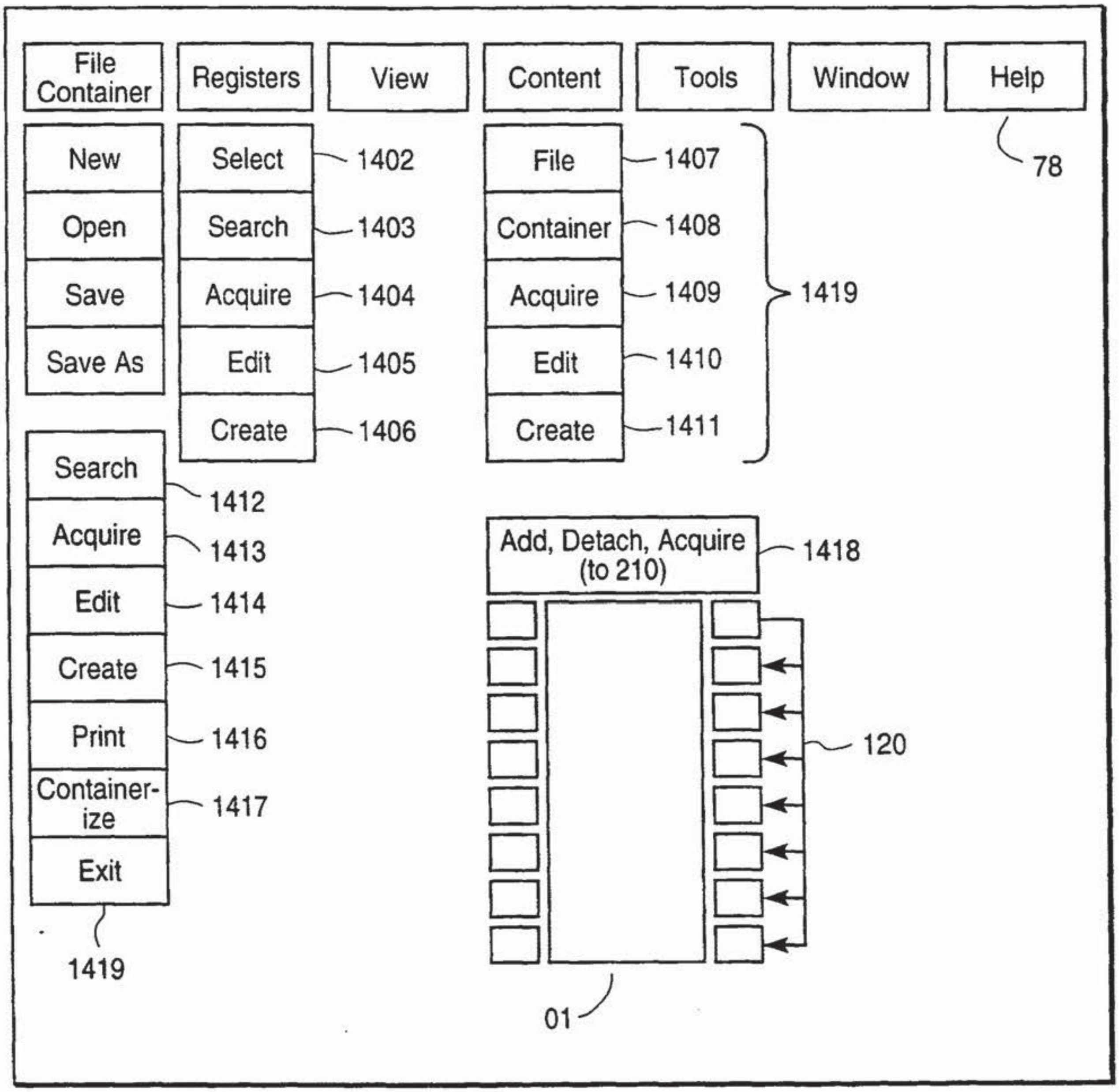
File Container
Registers
View
Content
Tools
Window
Help
78
New
Open
Save
Save As
Select
1402
Search
1403
Acquire
1404
Edit
1405
Create
1406
File
1407
Container
1408
Acquire
1409
Edit
1410
Create
1411
1419
Search
1412
Acquire
1413
Edit
1414
Create
1415
Print
1416
Container-ize
1417
Exit
1419
Add, Detach, Acquire (to 210)
1418
120
01

FIG. 14

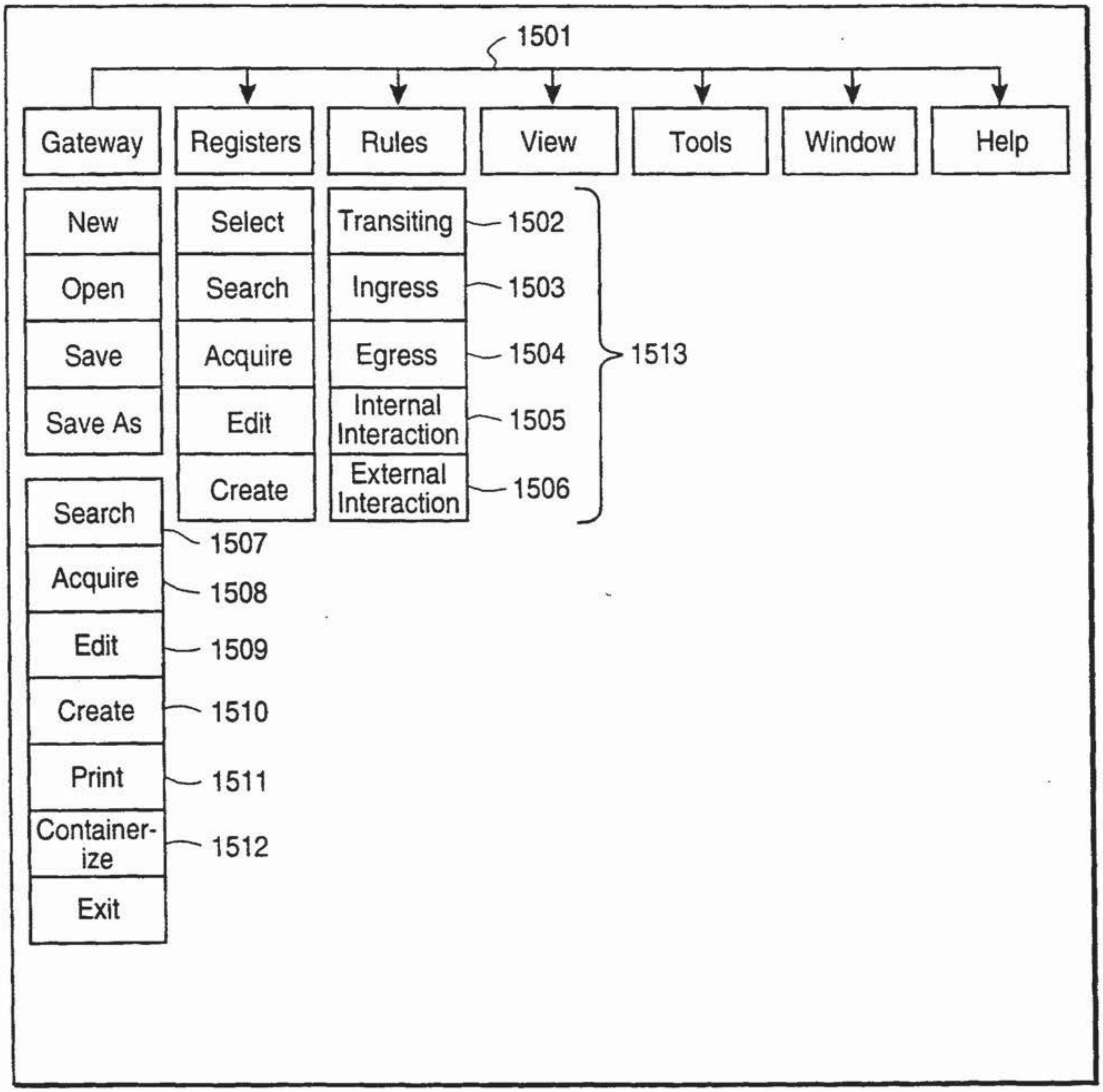
1501
Gateway
Registers
Rules
View
Tools
Window
Help
New
Open
Save
Save As
Select
Search
Acquire
Edit
Create
Transiting
1502
Ingress
1503
Egress
1504
Internal Interaction
1505
External Interaction
1506
1513
Search
1507
Acquire
1508
Edit
1509
Create
1510
Print
1511
Container-ize
1512
Exit

FIG. 15

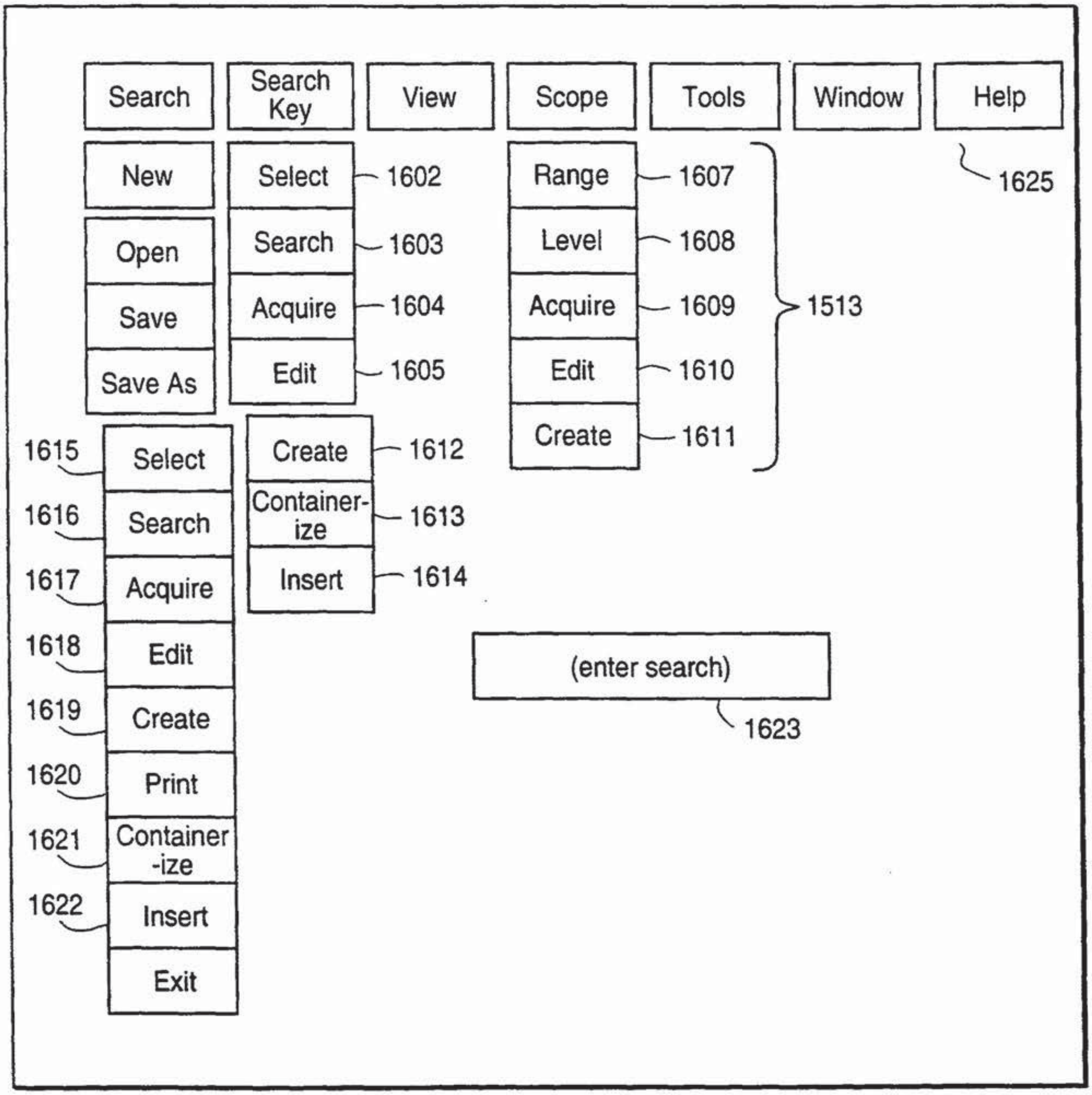
Search
Search Key
View
Scope
Tools
Window
Help
New
Open
Save
Save As
Select
1602
Search
1603
Acquire
1604
Edit
1605
Range
1607
Level
1608
Acquire
1609
Edit
1610
Create
1611
1513
1625
1615
Select
1616
Search
1617
Acquire
1618
Edit
1619
Create
1620
Print
1621
Container -ize
1622
Insert
Exit
Create
1612
Container-ize
1613
Insert
1614
(enter search)
1623

FIG. 16

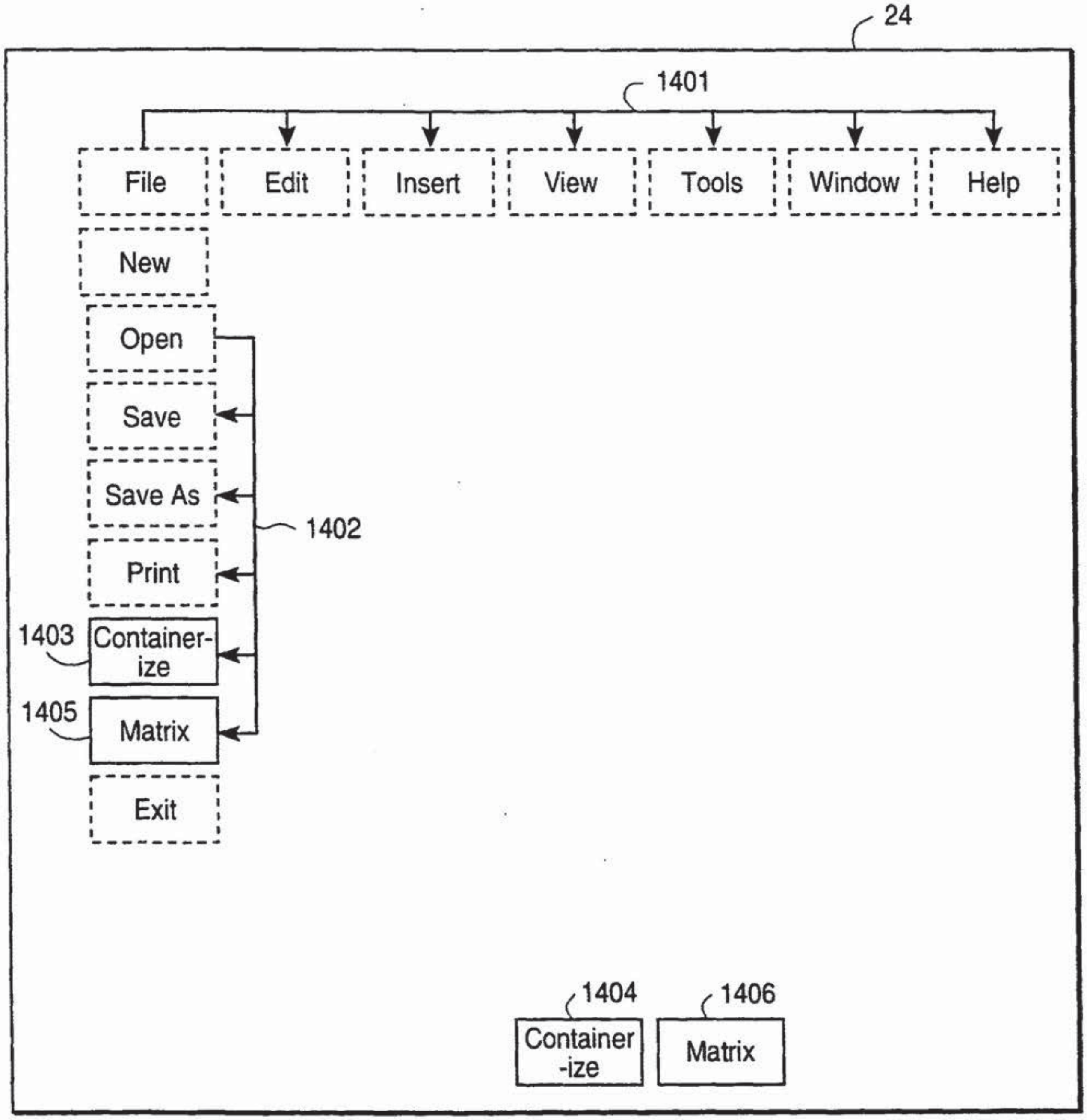
24
1401
File
Edit
Insert
View
Tools
Window
Help
New
Open
Save
Save As
1402
Print
1403
Container-ize
1405
Matrix
Exit
1404
Container-ize
1406
Matrix

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:	"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
"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	"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
"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)	"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
"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	"&"	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 09/284,113	**FILING OR 371(c) 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** *************************

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** CA	**SHEETS DRAWING** 30	**TOTAL CLAIMS** 36	**INDEPENDENT CLAIMS** 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