02-01-01 A

SKJERVEN MORRILL MACPHERSON LLP

Docket No.: M-7822 US

January 31, 2001

Box Patent Application
Commissioner For Patents
Washington, D. C. 20231

Enclosed herewith for filing is a patent application, as follows:

Inventor(s): Gilpin, Kevin E.; Stein, Adam R.

Title: Rule Based Configuration Engine For A Database

X	Return Receipt Postcard
X	This Transmittal Letter (in duplicate)
21	page(s) Specification (not including claims)
11	page(s) Claims
1	page Abstract
7	Sheet(s) of Drawings
3	page(s) Declaration For Patent Application and Power of Attorney
1	page NonPublication Request
1	page(s) Recordation Form Cover Sheet (in duplicate)
2	page(s) Assignment

CLAIMS AS FILED

For	Number Filed			Number Extra		Rate			Basic Fee
								$	710.00
Total Claims	76	-20	=	56	x	$ 18.00	=	$	1,008.00
Independent Claims	8	-3	=	5	x	$80	=	$	400.00
☐ Fee of _____ for the first filing of one or more multiple dependent claims per application								$	

Please make the following charges to Deposit Account 19-2386:

☒ Total fee for filing the patent application in the amount of $ 2,118.00
☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account 19-2386.

EXPRESS MAIL LABEL NO:
EL764880543US

Respectfully submitted,

Mary Jo Bertani

Mary Jo Bertani
Attorney for Applicant(s)
Reg. No. 42,321

25 Metro Drive • Suite 700 • San Jose, CA 95110 • Phone (408) 453-9200 • Fax (408) 453-7979
San Francisco, CA • Austin, TX • Newport Beach, CA

719578 v1

EXPRESS MAIL LABEL NO:
EL764880540US

MODIFIED PTO/SB/35 (11-00)

REQUEST AND CERTIFICATION UNDER 35 U.S.C. 122(b)(2)(B)(i)		
	Inventors	Gilpin, Kevin E.; Stein, Adam R.
	Title	Rule Based Configuration Engine For A Database
	Atty Docket Number	M-7822 US

I hereby certify that the invention disclosed in the attached application **has not and will not be** the subject of an application filed in another country, or under a multilateral agreement, that requires publication at eighteen months after filing. I hereby request that the attached application not be published under 35 U.S.C. 122(b).

January 31, 2001
Date

Mary Jo Bertani
Attorney for Applicants
Reg. No.: 42,321

This request must be signed in compliance with 37 CFR 1.33(b) and submitted with the application **upon filing.**

Applicant may rescind this nonpublication request at any time. If applicant rescinds a request that an application not be published under 35 U.S.C. 122(b), the application will be scheduled for publication at eighteen months from the earliest claimed filing date for which a benefit is claimed.

If applicant subsequently files an application directed to the invention disclosed in the attached application in another country, or under a multilateral international agreement, that requires publication of applications eighteen months after filing, the applicant **must** notify the United States Patent and Trademark Office of such filing within forty-five (45) days after the date of the filing of such foreign or international application. **Failure to do so will result in abandonment of this application (35 U.S.C. 122(b)(2)(B)(iii)).**

7 CFR 1.213(a) provides for a request that an application not be published under 35 U.S C. 122(b). Confidentiality is governed by 35 U S.C 122 and 37 CFR 1 14. SEND TO. Commissioner for Patents, Washington, DC 20231.

"Express Mail" mailing label number:

EL764880543US

RULE BASED CONFIGURATION ENGINE FOR A DATABASE

Kevin E. Gilpin
Adam R. Stein

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates generally to computerized configuring systems. More specifically, this invention relates to a system and method for testing the compatibility of parts included in a configuration.

Description of the Related Art

Many products are comprised of individual parts or components. Currently, configuring systems, also referred to as configuration engines, are available that allow a user to configure a product by interactively selecting components from among various groups based on availability and compatibility of features and options for the product. One known system is described in U.S. Patent No. 5,825,651, entitled "Method and Apparatus For Maintaining and Configuring Systems," issued October 20, 1998, (hereinafter the "'651 patent"), which is assigned to the same assignee as the present invention, and is hereby incorporated by reference.

In one embodiment of a configuration system disclosed in the '651 patent, a framework for defining a product line includes a set of related components that are selected from a parts catalog. A product might consist of several hundred individual parts that are organized into part groups and are available on one or more of a number of products. A product is modeled by describing which parts and part groups are available in that product and which choices must be made from within the part groups, and then by writing additional rules that describe part-to-part relationships which are not modeled by the product structure. A compiler converts the product

structure and the rules into four rule types: includes (parts that are included by default), excludes, removes, and requires choice (a choice among a group of parts that must be made to achieve a valid configuration). Parts may also be classified as optional which signifies that they can be optionally included in the configuration.

After compilation, there may be several hundred, several thousand, or even more of these rules. When the system loads the model, all parts and products should initially be in a "selectable" state, which means that the client or user is allowed to choose them. When the client selects a part, that part is put in the "selected" state. Parts that are included by the selected parts enter the "included" state, and parts that are excluded by the selected parts enter the "excluded" state. Parts that were previously included but are removed by a "removes" rule are in the "deleted" state. Each part must always be in exactly one state. Parts that are selected by a user or are included are referred to as "selected". Parts that are excluded or deleted are referred to as "not selectable".

As product models grow in size and complexity, configuration errors may occur when a rule or series of rules is not properly defined and produces an undesired effect, such as the exclusion of a part that should be selectable. Configuration errors may also occur when a series of improperly defined rules causes a part to be in more than one state at the same time, such as "included" and "excluded", or "selected" and "deleted".

For large models, such errors may be difficult to find due to the large number of rules in the model, the unexpected effects of some configuration operations, and the complex interactions between rules. It is therefore desirable to have an automated testing tool to locate and analyze configuration errors, so that the rules may be corrected.

SUMMARY

The invention provides in one embodiment the ability to test rules in a rule-based system for configuring a product. A configuration system defines the

components of a product using elements contained in a parts catalog and rules that define relationships between the components of a product. The user provides test cases that select at least one part to include in the product configuration, and the configuration tester processes the rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

In one embodiment, the invention provides a method of testing a product configuration in a system for generating product configurations that include a variety of component parts. The configuration system includes one or more rules that define a relationship between at least two parts. The method includes entering a test case that selects at least one part to include in the product configuration. The system then processes the rule to determine whether part selected in the test case conflicts with parts that are already included in the product configuration, that is, whether the rule conflicts with other rules.

To test new rules, one embodiment of the invention initializes the configuration system with a part state and inputs at least one part selection as specified in the test case. A component referred to as a "listener" detects state change events that result in the system being in the initialized part state. Another component of the invention generates a cause that explains the part state in terms of the state change event, and generates a new part state for each part associated with the cause. The invention then determines the cause or causes that explain the new part states in terms of the state change event.

One feature of an embodiment of the invention generates a cause tree wherein the root of the cause tree is the initial part state, and "leaves" of the tree are the user's selections of parts.

Another component of an embodiment of the invention is an "explainer" which generates an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections. The explanation(s) are based on selection of a part, execution of a rule, a part being in two

states at the same time, a requires choice rule that cannot be satisfied, or on a look ahead process. To provide an explanation of how the system arrived at a particular part state, the invention sorts the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

In another embodiment, the invention is distributed as an article of manufacture, namely a computer usable medium having computer readable program code embodied therein for testing a product configuration in a system for generating product configurations. The system includes at least one rule defining a relationship between at least two parts, and the product configuration includes a plurality of parts.

The computer readable program code is configured to cause a computer to allow a user to enter a test case, wherein the test case selects at least one part to include in the product configuration. The program code then causes a computer to process the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration. This is accomplished by the computer readable program code causing a computer to initialize the system with a part state, to input the part selection to the system; and to listen to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

The program code then causes a computer system to determine the cause or causes that explain the new part states in terms of the state change event.

One feature of the program code causes a computer to generate a cause tree wherein the root of the cause tree is the initial part state, and "leaves" of the tree are the user's selections of parts.

Another component of the program code causes a computer to execute a component referred to as an "explainer" which generates an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections. The explanation(s) are based on selection of a part,

execution of a rule, a part being in two states at the same time, a requires choice rule that cannot be satisfied, or on a look ahead process. To provide an explanation of how the system arrived at a particular part state, the invention sorts the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

The foregoing has outlined rather broadly the objects, features, and technical advantages of the present invention so that the detailed description of the invention that follows may be better understood.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system with which the present invention may be utilized.

Figure 2 is a block diagram of an embodiment of a maintenance and configuration system with which the present invention may be utilized.

Figure 3 is a block diagram of a maintenance and configuration tester system according to an embodiment of the present invention.

Figure 3a is a block diagram of configuration tester modules included in an embodiment of the present invention.

Figure 3b is a diagram of an example of a cause/effect tree.

Figure 3c is a diagram of an example of a lookahead subtree embedded within a cause/effect tree.

Figure 3d is a diagram of an example of a lookahead subtree collapsed to a single node in the cause/effect tree.

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

DETAILED DESCRIPTION

A method and apparatus for testing a system for maintaining and configuring products is described. The present invention can be implemented on a general purpose computer system 130 such as illustrated in Fig. 1. Computer system 130 may be one of many workstations or servers connected to a network such as a local area network (LAN), a wide area network (WAN), or a global information network such as the Internet through network interface 140.

CPU 132 can be constructed from one or more microprocessors and/or integrated circuits. Main memory 136 stores programs and data that CPU 132 may access. When computer system 130 starts up, an operating system program is loaded into main memory 136. The operating system manages the resources of computer system 130, such as CPU 132, audio controller 142, storage device controller 138, network interface 140, I/O controllers 146, and host bus 134. The operating system reads one or more configuration files to determine the hardware and software resources connected to computer system 130.

During operation, main memory 136 includes the operating system, configuration file, and one or more application programs with related program data. Application programs can run with program data as input, and output their results as program data in main memory 136 or to one or more mass storage devices through a memory controller (not shown) and storage device controller 138. CPU 132 executes one or more application programs, including one or more programs to establish a connection to a computer network through network interface 140. The application programs may be embodied in one executable module or may be a collection of routines that are executed as required. Operating systems commonly generate "windows", as well known in the art, to present information about or from an

application program, and/or to allow a user to interact with an application program. Each application program typically has its own window that is generated when the application program is executing. Each window may be minimized to an icon, maximized to fill the display, overlaid in front of other windows, and underlaid behind other windows.

Storage device controller 138 allows computer system 130 to retrieve and store data from mass storage devices such as magnetic disks (hard disks, diskettes), and optical disks (DVD and CD-ROM). The information from the DASD can be in many forms including application programs and program data. Data retrieved through storage device controller 138 is usually placed in main memory 136 where CPU 132 can process it.

One skilled in the art will recognize that the foregoing components and devices are used as examples for sake of conceptual clarity and that various configuration modifications are common. For example, audio controller 142 is connected to PCI bus 156 in Fig. 1a, but may be connected to the ISA bus 138 or reside on the motherboard (not shown) in alternative embodiments. As further example, although computer system 130 is shown to contain only a single main CPU 132 and a single system bus 134, those skilled in the art will appreciate that the present invention may be practiced using a computer system that has multiple CPUs 132 and/or multiple busses 134. In addition, the interfaces that are used in the preferred embodiment may include separate, fully programmed microprocessors that are used to off-load computationally intensive processing from CPU 132, or may include input/output (I/O) adapters to perform similar functions. Further, PCI bus 156 is used as an exemplar of any input-output devices attached to any I/O bus, AGP bus 159 is used as an exemplar of any graphics bus; graphics device 154 is used as an exemplar of any graphics controller; and host-to-PCI bridge 160 and PCI-to-ISA bridge 162 are used as exemplars of any type of bridge. Consequently, as used herein the specific exemplars set forth in Fig. 1 are intended to be representative of their more general classes. In general, use of any specific exemplar herein is also intended to be representative of its class and the non-inclusion of such specific devices in the foregoing list should not be taken as indicating that limitation is desired.

The invention detects and analyzes configuration errors in a system for configuring products such as described in the '651 patent. A brief description of the '651 patent is provided in the following paragraphs as background for understanding the present invention.

Brief Description Of The '651 Patent

Referring to Fig. 2, one embodiment of configuration engine 200 disclosed in the '651 patent is shown. Configuration engine 200 is rule-based, and includes maintenance environment 202 and configuration environment 204. Maintenance environment 202 includes zero or more individual parts, or components, in parts catalog 206. Part relationships 208 defines relationships between a first set of parts and a second set of parts so that when all of the members of the first set of parts are selected, the relationship between the two sets is enforced on the parts in the second set. A set of parts can include multiple parts. The incorporation of parts in a set can be arbitrary. That is, a multi-part set can contain parts that are otherwise unrelated. For example, for the purpose of configuring an automobile, a set can contain parts such as an engine, sun roof, and a color. These parts seem to be unrelated, but they can be combined into a part relationship 208 for purposes of forming a relationship using an embodiment of configuration engine 200.

In one embodiment, there are four kinds of part relationships 208 between parts: requires choice, includes, excluded, and removes. An included part is included automatically. A part is excluded from the configuration when its inclusion would result in an invalid configuration. A part may be removed when another part is added. Thus, when a first part exists in the configuration and a second part is added, the first part is removed from the configuration if there is a conflict. The requires choice relationship is used to allow a set of choices to be made from a group of parts. The number of parts chosen is limited to a valid bounds specification. The relations that are created between parts within a product are enforced only on that particular product. However, if some part relationships 208 are to be enforced on all products within a product line, then the relations are generated once and enforced for all products.

One or more product definitions 210 are generated by a population of component parts. Using configuration engine 200, a user can configure a product given product definitions 210 and part relationships 208 associated with product definitions 210. Configuration environment 204 accepts a configuration user's input and processes it in product specification/verification 212 to verify the product configuration, and to update the specification based on the user's input, or to notify the user that the input is invalid based on product definitions 210 and user selections.

A graphical user interface (GUI) is used to allow the user to interactively generate product definitions 210. GUI operations such as drag, drop, and selection operations can be used to specify product definitions 210.

Relationships associated with items contained in product definitions 210 are evaluated when user input is received. Configuration engine 200 determines which relationships are active and inactive given the user input. A relationship is active when all the items in a product's product definition 210 are selected. A relationship is inactive until all of the parts in a product's product definition 210 are selected.

Configuration engine 200 is used to configure a product using a definition created by the maintenance environment 202. Configuration environment 204 ensures that the current configuration state is always valid. The user can select and unselect parts in any order. When user input is received, product specification/verification 212 validates the input based on the current state of the configuration. In addition, the product specification/verification 212 identifies selections that could cause a valid configuration to become invalid. Product specification/verification 212 removes these selections from the set of possible selections so that the user does not make an invalid selection.

Configuration engine 204 evaluates the current state of a configuration based on product definitions 210, part relationships 208, and state information. After receipt of input from a user, product specification/verification 212 evaluates relationships in both the forward and backward direction. Forward and backward evaluations can result in the addition or deletion of elements from the product configuration.

During configuration, information is stored in tables and vectors. Configuration engine 200 represents elements in a configuration (e.g., product, part, and group) as bits in a bit vector. Thus, for example, a vector includes a number of bits is equal to the total number of elements. An element's bit can be set or reset to specify the state of the element in the current configuration. For example, a user vector can be used that specifies for each element whether the element has been selected by the user during the configuration. In addition, excluded and removed vectors identify whether an element is excluded or removed (respectively) from a configuration. Vectors can be used to identify whether an element 1) has been selected (by the user or the configuration system), 2) is selectable, and 3) not selectable.

Tables contain element relationships. A table is used to represent the includes, excludes, removes, and requires choice relationships, for example. Each table has a left-hand side and a right-hand side that corresponds to the left-hand and right-hand sides of a relationship. In each case, the left-hand side is a bit vector that contains bits corresponding to elements. The includes, excludes and removes tables contain a bit vector in the right-hand side that represents configuration elements. The right-hand side of the requires choice table is a pointer that points to an entry in a group table. The group table entry is a bit vector that identifies the elements that are contained in the group from which a choice is to be made. The right-hand side of a requires choice table entry further includes minimum and maximum designations. Minimum and maximum values identify the minimum and maximum number of group members that are to be selected to satisfy a requires group relationship.

The bit vector implementation of relationships and internal runtime state allows for fast and efficient computation of relationship-based configuration. A comparison of bits can be performed in one machine instruction in most cases.

There are many ways that errors can be introduced into a configuration, however, the effects of these errors can be categorized in 2 groups:

1) A part is put in a state which was not intended by the user (state error), or

2) A part is put in more than one state at the same time (exception error).

Errors may be caused by a single rule, or by a chain of rules. Complex errors are often caused by a "look ahead" process included in product specification/verification 212 that test-selects each product (if more than one product is selectable) to make sure that it is in fact selectable. The look-ahead process helps insure that the state of a product is not reported as selectable when selection of that product would lead to a rule conflict. The look-ahead process also determines the sets of parts that are excluded or deleted by every selectable product.

Further errors may arise with requires choice rules, which typically require that between some minimum (*min*) and maximum (*max*) number of choices must be made from a set of parts. For example, there is always an implicit requires choice rule that specifies that at least exactly one (min/max = 1/1) part must be selected for a product. Requires choice rules are complex to evaluate because they may cause many kinds of inferences. In general, there is no way to determine whether a selectable part is actually selectable without selecting it and checking to see whether it causes a conflict. In order to ensure that each selectable part is not going to cause such a conflict, configuration engine 200 would have to select a selectable part after each user selection, which is too computationally expensive. The following examples of each type of error illustrate the problem.

State Errors

The simplest types of state errors are caused when a rule has been accidentally omitted or written. For example, the user may select PartA and PartB, and then note that 'PartC' is excluded rather than selectable. In the simplest case, this may be due to the following rule in the model:

PartA Excludes PartC

Or, there may be a rule:

PartA Requires Choice (min/max = 1/1) { PartB, PartC }

Here, selecting PartA implies that either PartB or PartC must be selected. Selecting PartB causes configuration engine 200 to infer that PartC must be Excluded. Alternatively, multiple rules may cause a state change, for example:

PartA Includes PartX
PartX Excludes PartC

Here, selecting PartA causes PartX to be included, which then causes PartC to be excluded.

State errors can also be caused by the look-ahead process. Suppose the following rules are written:

ProductA Excludes PartA
ProductB Includes PartB
ProductB, PartB Excludes PartA
ProductC RequiresChoice (min/max 1/1) PartA, PartC
ProductC Includes PartC

Even if the user has not made any selections, PartA will be excluded by the look ahead process for each of products A, B, and C. Detecting state changes that are caused by the look-ahead process is particularly difficult because for each product there may be a different rule chain or exception error that causes the state error.

Exception Errors

Sometimes, rules may be inadvertently written that cause a conflicting state exception. The simplest case can be summed up by the rules:

PartA includes PartB

PartB excludes PartA

If PartA is selected, then PartB will be Included. But then the second rule causes PartA to be excluded. This conflicting state cannot be reconciled, and an exception is raised.

Most exception conditions are more complex than this one. For example, selecting a part that is in a requires choice rule may cause the requires choice rule to be unsatisfiable as follows:

PartA requires choice (min/max=1/1) { PartB, Part C }
PartB includes PartC

In the preceding rules, if PartA is selected, selecting PartB will cause an exception error because the requires choice rule will not be satisfiable.

Configuration Testing

Fig. 3 shows an embodiment of the present invention for configuration tester system 300 that includes several components for detecting and analyzing configuration errors. One component is configuration tester graphical user interface (CTGUI) 302 that enables users to enter new rules 304 and define test cases 306 that describe the expected behavior of their models to test the configuration. New rules 304 are input to parts relationships 308 and product definitions 310 in configuration engine 312. Test cases 306 describe one or more sets of selections that should be made, and sets of parts and their expected states based on new rules 304, as well as rules previously included in parts relationships 308 and product definitions 310. For example, test cases 306 may describe the selection of a product and several parts. It may then ensure that some other set of parts is excluded, and a third set is included. An example of a test case in test cases 306 is:

Select ProductA
Select PartA
Ensure that (PartB, PartC) are excluded

Ensure that (PartD) is included

Once test cases 306 are written, configuration tester modules 314 run each test case 306 and verify that the tested parts are in the right state. If a test fails, configuration tester modules 314 determine why a part is in a certain state and explain the state as described below. The database of pre-existing rules can then be modified to correct errors found by configuration tester modules 314.

Configuration tester modules 314 include driver and listener module 316, debug engine 318, and explainer 320. Fig. 3a shows interrelationships of configuration tester modules 314 including types of data communicated between driver and listener 316, debug engine 318, and explainer 320, during operation. Driver and listener 316 selects parts from test cases 306 and sends the part selections to debug engine 318.

Debug engine 318 processes new rules 304 using the part selections, and sends state change events to driver and listener 316 as state changes result from the rules executing, exceptions occurring, and execution of the look ahead process. In the '651 patent, configuration engine 200 (Fig. 2) is optimized for very high performance. In one embodiment, configuration tester system 300 includes components of configuration engine 200 such as parts catalog 206, parts relationships 208, product definitions 210, and product specification/verification 210. Configuration tester system 300 can run in test mode or normal mode so that no performance penalties are imposed when operating configuration tester system 300 in normal mode. This is accomplished by executing all features and components of configuration tester system 300 from debug engine 318, which is only used in test mode.

The application program interface (API) to debug engine 318 includes program instructions to include new rules 304 with existing rules in parts relationships 208 and product definitions 210, and to run test cases 306 through product specification/verification 210. Debug engine 318 presents the same API as the normal mode of configuration engine 200 for selecting parts. CTGUI 302 is used to specify which test cases to run. Whenever a condition occurs that causes a part state

change, debug engine 318 detects this condition and transmits an appropriate notice to driver and listener 316 for the listener portion to handle and interpret the events.

Driver and listener 316 listens to the state change events and constructs a tree of the rule chains that are executing in debug engine 318 and resulting states. When a state error occurs, driver and listener 316 executes a driver to recreate the error condition for the part for which the state error occurred, along with all the part selections that caused the error to occur. The combination of the part and its state is represented by a part state.

In one embodiment, the part-state includes an identifier for the part, the state of the part, the selections which have been made (which are always a subset of the total user selections), and, optionally, the product for which lookahead is currently being run. For example, a part-state may represent:

Part A is included after selecting Part X and Part Y,
or
Part B is excluded with no selections during lookahead on Product X.

Each part-state also has a Cause, which is initially null. Configuration tester system 300 determines the Cause of the part state (a rule firing, an exception, a user selection, etc) and sets the Cause of the part-state.

Driver and listener 316 interfaces with debug engine 318. The driver portion of driver and listener 316 starts submitting the part selections that led to the error until a state change event occurs that recreates the error condition. The listener portion of driver and listener 316 is responsible for handling the state change events. It may happen as a result of any of the following:

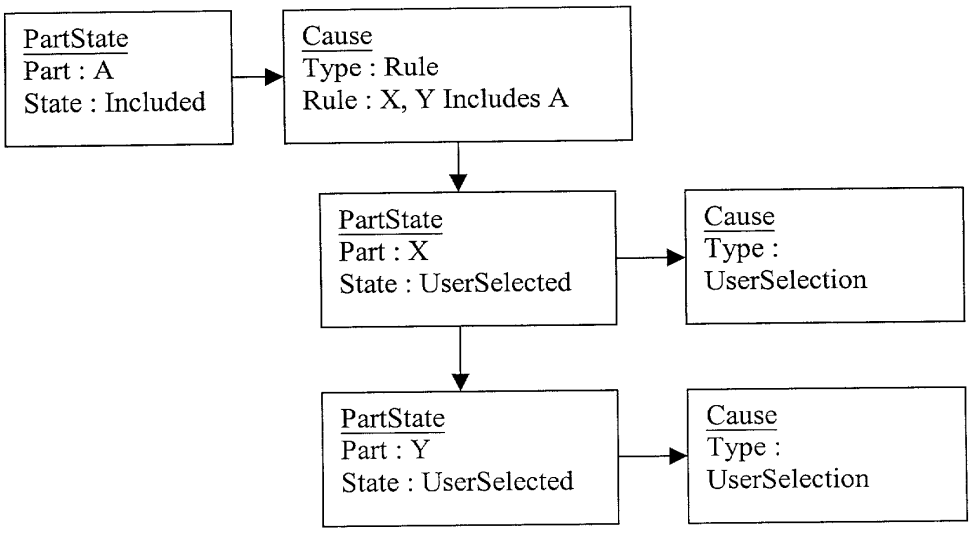
1) A user selection
2) A rule executing
3) A rule conflict (exception error)
4) Operation of the look ahead process

In each case, the driver generates a cause, which represents the event and the state change that resulted from it. Then, based on this new information, further analysis to explain the part state may be required to explain the error in accordance with the following summary:

Cause	Explanation Complete?	Next steps
User Selection	Yes	
Rule Executing	No	Determine why the rule executed
Conflicting State Exception (part is in 2 states at the same time)	No	Explain each of the conflicting states
Unsatisfiable requires choice exception	No	Determine why the requires choice rule executed, and explain the state of the parts that caused it to be unsatisfiable
Look ahead process	No	Explain the state of the part in each selectable product

Driver and listener 316, and debug engine 318, are recursive. The driver portion of driver and listener 316 is initialized with a single part state, along with a set of user selections. The user selections are specified in the test case. The driver inputs each user selection one by one, until the listener gets a state change event that explains the part state. Then the listener generates a cause that explains the part state in terms of the event. The listener also generates a new part state for each part associated with the cause. Then driver and listener 316 start over to find the causes that explain the new part states. Eventually, all part states can be explained in terms of a user selection. The explanation of the original part state is thus represented by a tree of causes and part states. The original part state is the root of the tree. The second level of the tree, i.e., the leaves, consist of the causes that caused the root part state. The next level is the part states that caused the causes, and so on.

For example, in one embodiment, suppose the task is: Explain why A is Included if X and Y are UserSelected. The tree might look like this:



Each PartState points to its Cause. If the Cause is a RuleCause, the Cause points to the parts that caused the rules to fire and their state is in turn explained with Cause objects.

Explainer 320 converts the tree into a format that readily allows the user to visualize the rules that are causing an erroneous result in the configured product. The root of the tree is the initial goal part state, and the leaves of the tree are the user's selections of parts. It is more intuitive to the user, however, to see the part selections as the root of the explanation, and then the chain of causes that follow from these selections. Accordingly, explainer 320 accepts the tree as input, and generates a description of the sequence of events by modeling the logical operation of configuration engine 312, not the literal sequence of actions. This is because converting the tree requires more than post-order traversal, which only provides a trace of the state of configuration engine 312. Logically, configuration engine 312 operates in a series of cause-and-effects iterations. In each iteration, configuration engine 312 first determines which rules should execute, and then applies the results of those rules to the current state of the configuration. The process then repeats until the

internal state of the configuration is no longer changing with each iteration. At this point, equilibrium is reached, and configuration engine 312 is ready to once again receive another selection of a part as input.

Explainer 320 determines the stem for each cause in a given iteration from part states in previous iterations, and determines the cause for each part state in the same iteration. This provides a mechanism for grouping and sorting the tree by iteration. In the simple case, the iteration number of a given part state is determined by measuring the longest distance between a part state and a leaf cause. For any given node in the cause/effect tree, the distances between that node and all the leaves of the tree that connect to that node can be counted. The maximum of this set of values is the maximum depth of the node, which is also the iteration number for that cause/effect. Fig. 3b shows an example of a cause/effect tree where the maximum depth of cause/effect 4 is two (2) (level 3 minus level 1).

Consider, for example, the following set of rules:

1) A includes X
2) B excludes Y
3) A,C,X require Y,Z

And the following sequence of events:

1) User picks A
2) Rule 1 brings in part X
3) User picks B
4) Rule 2 excludes Y
5) User picks C
6) Rule 3 includes Z

There are several things to notice in this example. First, the order of user selections is irrelevant with regard to the logical operation of configuration engine 312 is concerned. Also, the order of execution of rules 1 and 2 is irrelevant. These details are abstracted away when the sequence of events is broken into logical rounds:

Round 1: User selects A, B, C
Round 2: Rule 1 includes X
Rule 2 excludes Y
Round 3: Rule 3 includes Z

The latter description eases understanding the logical flow of configuration engine 312, and better highlights the dependencies between user actions and rules. This is especially true in situations involving more complex series of rules. For the preceding example, the latter representation makes it immediately clear that the activations of rules 1 and 2 are not causally linked events, whereas the first representation leaves open the possibility that rule 2 executes as a consequence of rule 1.

Complications Caused By Look Ahead

In the look ahead process, configuration engine 312 makes a series of selections to determine what would happen if the user chose particular parts. Many rules can execute within a particular look ahead scenario, but eventually all of these rule executions are retracted, and the results of the look ahead process are applied to the current product being configured. Therefore, an entire look ahead event happens within an individual round of configuration engine 312 activity, even though the look ahead event itself may contain many rounds of executing rules. The recursive aspect of the causes and part states tree is taken into account to invert the causes and part states tree with explainer 320. Essentially, explainer 320 regards look ahead events as branches within the main tree, and collapses them down to single nodes when calculating the proper round in which to place a given cause or part state. An example of what happens during look ahead is: given two products, P1 and P2, and the rules 'P1 excludes A', 'P2 excludes A', Lookahead internally selects each selectable product in turn, and determines whether there are any parts which are excluded by all products. In this example, A would be excluded by lookahead. To the explainer, this can be summarized as 'A is excluded by Lookahead', but within each product, the rules provide a further cause.

Figs. 3c and 3d show how lookahead nodes are collapsed to a single node of the main cause/effect tree. Specifically, Fig. 3c shows lookahead nodes 4.1 though 4.5 expanded within the main cause/effect tree 322, while Fig. 3d shows lookahead subtree 324 collapsed into lookahead cause/effect 4 in main cause/effect tree 326.

In one embodiment, Explainer 320 is designed in an object-oriented fashion that allows key elements of the process to be overridden to provide specialized behavior. For example, some configuration models are generated automatically from known product data descriptions or other sources. Explainer 320 can be overridden to trace explanations all the way back to these original rule sources. Explainer 320 can also be overridden to integrate data from historical logs or databases, as well as data input by the user.

The present invention has been described in the context of a fully functional computer system, however those skilled in the art will appreciate that the present invention is capable of being distributed as a program product in a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of signal bearing media include: recordable type media such as floppy disks and CD-ROM, transmission type media such as digital and analog communications links, as well as media storage and distribution systems developed in the future.

Additionally, the foregoing detailed description has set forth various embodiments of the present invention via the use of block diagrams, flowcharts, and examples. It will be understood by those within the art that each block diagram component, flowchart step, and operations and/or components illustrated by the use of examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof. In one embodiment, the present invention may be implemented via Application Specific Integrated Circuits (ASICs). However, those skilled in the art will recognize that the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard integrated circuits, as a computer program running on a computer, as firmware, or as virtually any combination thereof and that designing the circuitry and/or writing the

code for the software or firmware would be well within the skill of one of ordinary skill in the art in light of this disclosure.

While the invention has been described with respect to the embodiments and variations set forth above, these embodiments and variations are illustrative and the invention is not to be considered limited in scope to these embodiments and variations. Accordingly, various other embodiments and modifications and improvements not described herein may be within the spirit and scope of the present invention, as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method of testing a product configuration in a system for generating product configurations, the system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the method comprising:

entering a test case, wherein the test case selects at least one part to include in the product configuration; and

processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

2. The method, as set forth in claim 1, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

initializing the system with a part state;

inputting the at least one part selection; and

listening to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

3. The method, as set forth in claim 2, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

generating a cause that explains the part state in terms of the state change event.

4. The method, as set forth in claim 3, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

generating a new part state for each part associated with the cause.

5. The method, as set forth in claim 4, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

determining the causes that explain the new part states in terms of the state change event.

6. The method, as set forth in claim 5, further comprising:

generating a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

7. The method, as set forth in claim 6, further comprising:

generating an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

8. The method, as set forth in claim 7, wherein the explanation is based on selection of a part.

9. The method, as set forth in claim 7, wherein the explanation is based on execution of a rule.

10. The method, as set forth in claim 7, wherein the explanation is based on a part being in two states at the same time.

11. The method, as set forth in claim 7, wherein the explanation is based on a requires choice rule that cannot be satisfied.

12. The method, as set forth in claim 7, wherein the explanation is based on a look ahead process.

13. The method, as set forth in claim 7, further comprising:

sorting the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

14. An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for testing a product configuration in a system for generating product configurations, the system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the computer readable program code including:

computer readable program code configured to cause a computer to allow a user to enter a test case, wherein the test case selects at least one part to include in the product configuration; and

computer readable program code configured to cause a computer to process the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

15. The article of manufacture, as set forth in claim 14, further including:

computer readable program code configured to cause a computer to initialize the system with a part state;

computer readable program code configured to cause a computer to input the at least one part selection; and

computer readable program code configured to cause a computer to listen to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

16. The article of manufacture, as set forth in claim 15, further including:

computer readable program code configured to cause a computer to generate a cause that explains the part state in terms of the state change event.

17. The article of manufacture, as set forth in claim 16, further including:

computer readable program code configured to cause a computer to generate a new part state for each part associated with the cause.

18. The article of manufacture, as set forth in claim 17, further including:

computer readable program code configured to cause a computer to determine the causes that explain the new part states in terms of the state change event.

19. The article of manufacture, as set forth in claim 18, further comprising:
computer readable program code configured to cause a computer to generate a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

20. The article of manufacture, as set forth in claim 19, further comprising:
computer readable program code configured to cause a computer to generate an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

21. The article of manufacture, as set forth in claim 20, wherein the explanation is based on selection of a part.

22. The article of manufacture, as set forth in claim 20, wherein the explanation is based on execution of a rule.

23. The article of manufacture, as set forth in claim 20, wherein the explanation is based on a part being in two states at the same time.

24. The article of manufacture, as set forth in claim 20, wherein the explanation is based on a requires a choice rule that cannot be satisfied.

25. The article of manufacture, as set forth in claim 20, wherein the explanation is based on a look ahead process.

26. The article of manufacture, as set forth in claim 20, further comprising:
computer readable program code configured to cause a computer to sort the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

27. An apparatus for testing a product configuration generated by a product configuration system, comprising:

at least one rule defining a relationship between at least two parts in the product configuration;

a test case pertaining to at least one part to include in the product configuration; and

a processor coupled to receive the at least one rule and the test case, wherein the processor is operable to determine whether the at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to the at least one rule.

28. The apparatus, as set forth in claim 27, wherein the processor is further operable to:

initialize the configuration system with a part state;

to input the at least one part selection;

to listen to state change events in the system; and

to detect when a state change event occurs that results in the configuration system being in the initialized part state.

29. The apparatus, as set forth in claim 28, wherein the processor is further operable to:

generate a cause that explains the part state in terms of the state change event.

30. The apparatus, as set forth in claim 29, wherein the processor is further operable to:

generate a new part state for each part associated with the cause.

31. The apparatus, as set forth in claim 30, wherein the processor is further operable to:

generate a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

32. The apparatus, as set forth in claim 30, wherein the processor is further operable to:

generate an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

33. The apparatus, as set forth in claim 32, wherein the explanation is based on execution of a rule.

34. The apparatus, as set forth in claim 32, wherein the explanation is based on a part being in two states at the same time.

35. The apparatus, as set forth in claim 32, wherein the explanation is based on a requires a choice rule that cannot be satisfied.

36. The apparatus, as set forth in claim 32, wherein the explanation is based on a look ahead process.

37. The apparatus, as set forth in claim 30, wherein the processor is further operable to:

sort the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

38. A configuration system comprising:

a modified database, wherein the modified database is based on the results of testing a product selection using a test case, wherein the test case includes at least one part selection, and the expected state of the at least one selected part.

39. The configuration system of claim 38, wherein the test case further includes the product selection.

40. The configuration system of claim 38, further comprising:

at least one vector, wherein said vector comprises a bit field, further wherein the bit field comprises bits that represent elements in a configuration.

41. The configuration system of claim 40, wherein the number of bits in the bit field is equal to the total number of elements and an element's bit can be set or reset to specify the state of the element in the configuration.

42. The configuration system of claim 40, wherein the vector specifies whether an element has been selected by the user during the configuration.

43. The configuration system of claim 40, wherein excluded vectors identify whether an element is excluded from a configuration.

44. The configuration system of claim 40, wherein removed vectors identify whether an element is removed from a configuration.

45. The configuration system of claim 40, wherein the vector identifies whether an element is selectable.

46. A database comprising:

at least one table, wherein said table represents relationships between elements in a configuration; and

at lease one modified rule, wherein the rule is modified based on the results of testing a product selection.

47. The database of claim 46, wherein said table represents "includes" relationships between elements in a configuration.

48. The database of claim 46, wherein said table represents "excludes" relationships between elements in a configuration.

49. The database of claim 46, wherein said table represents "removes" relationships between elements in a configuration.

50. The database of claim 46, wherein said table represents "requires choice" relationships between elements in a configuration.

51. The database of claim 50, wherein the representation of "requires choice" relationships includes a pointer to a group table that includes a bit vector that identifies the elements that are contained in the group from which a choice is to be made.

52. The database of claim 50, wherein the representation of "requires choice" relationships includes minimum and maximum designations to identify the minimum and maximum number of group members that are to be selected to satisfy the "requires choice" relationship.

53. The database of claim 46, wherein said table includes a left-hand side and a right-hand side.

54. The database of claim 53, wherein the left-hand side includes a bit vector that contains bits corresponding to elements.

55. The database of claim 53, wherein the right-hand side includes one or more bit vectors that represent configuration elements.

56. A test case for testing a product configuration generated by a product configuration system, comprising:

a product selection;

at least one part selection; and

an expected state of the selected part based on one or more rules.

57. A method for identifying an invalid configuration generated by a product configuration system, comprising:

selecting a product;

selecting at least one part; and

generating a part state of the selected part based on one or more rules.

58. The method as set forth in claim 57, further comprising:

generating a cause that explains the part state in terms of a state change event.

59. The method as set forth in claim 57, further comprising:

detecting an error in the part state.

60. The method as set forth in claim 57, further comprising:

detecting when the at least one part is put in more than one state at the same time.

61. The method as set forth in claim 57, further comprising:

determining whether the product is selectable.

62. The method as set forth in claim 57, further comprising:

detecting when the at least one part is put in more than one state at the same time.

63. The method as set forth in claim 57, further comprising:

reporting the state of the product as not selectable when selection of the product would conflict with the rule.

64. The method as set forth in claim 57, further comprising:

determining sets of parts that are excluded or deleted based on the product.

65. The method as set forth in claim 57, further comprising:

detecting when a state change event occurs that results in the system being in the initialized part state.

66. The method as set forth in claim 65, further comprising:
generating a cause that explains the part state in terms of the state change event.

67. The method, as set forth in claim 66, further comprising:
generating a new part state for each part associated with the cause.

68. The method, as set forth in claim 67, further comprising:
generating a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

69. The method, as set forth in claim 68, further comprising:
generating an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

70. An apparatus for testing a product configuration generated by a product configuration system, comprising:
means for defining a relationship between at least two parts in the product configuration;
means for defining a test case for at least one part to include in the product configuration; and
means for determining whether the at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to at least one rule.

71. The apparatus, as set forth in claim 70, further comprising:
means for initializing the configuration system with a part state;
means for detecting a state change event in the system; and
means for detecting when a state change event occurs that results in the configuration system being in the initialized part state.

72. The apparatus, as set forth in claim 71, further comprising:

means for generating a cause that explains the part state in terms of the state change event.

73. The apparatus, as set forth in claim 72, further comprising:
means for generating a new part state for each part associated with the cause.

74. The apparatus, as set forth in claim 73, further comprising:
means for generating a cause tree, wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

75. The apparatus, as set forth in claim 73, further comprising:
means for generating an explanation of the part state, wherein the part selections are the root of the explanation and the causes follow from the part selections.

76. The apparatus, as set forth in claim 70, further comprising:
means for modifying the at least one rule when the test case conflicts with the plurality of parts previously included in the product configuration.

RULE BASED CONFIGURATION ENGINE FOR A DATABASE

Kevin E. Gilpin
Adam R. Stein

Abstract of the Disclosure

The invention provides the ability to test rules in a rule-based system for configuring a product. The configuration system defines the components of a product using elements contained in a parts catalog and rules that define relationships between the components of a product. The user provides test cases that select at least one part to include in the product configuration, and the configuration tester processes the rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

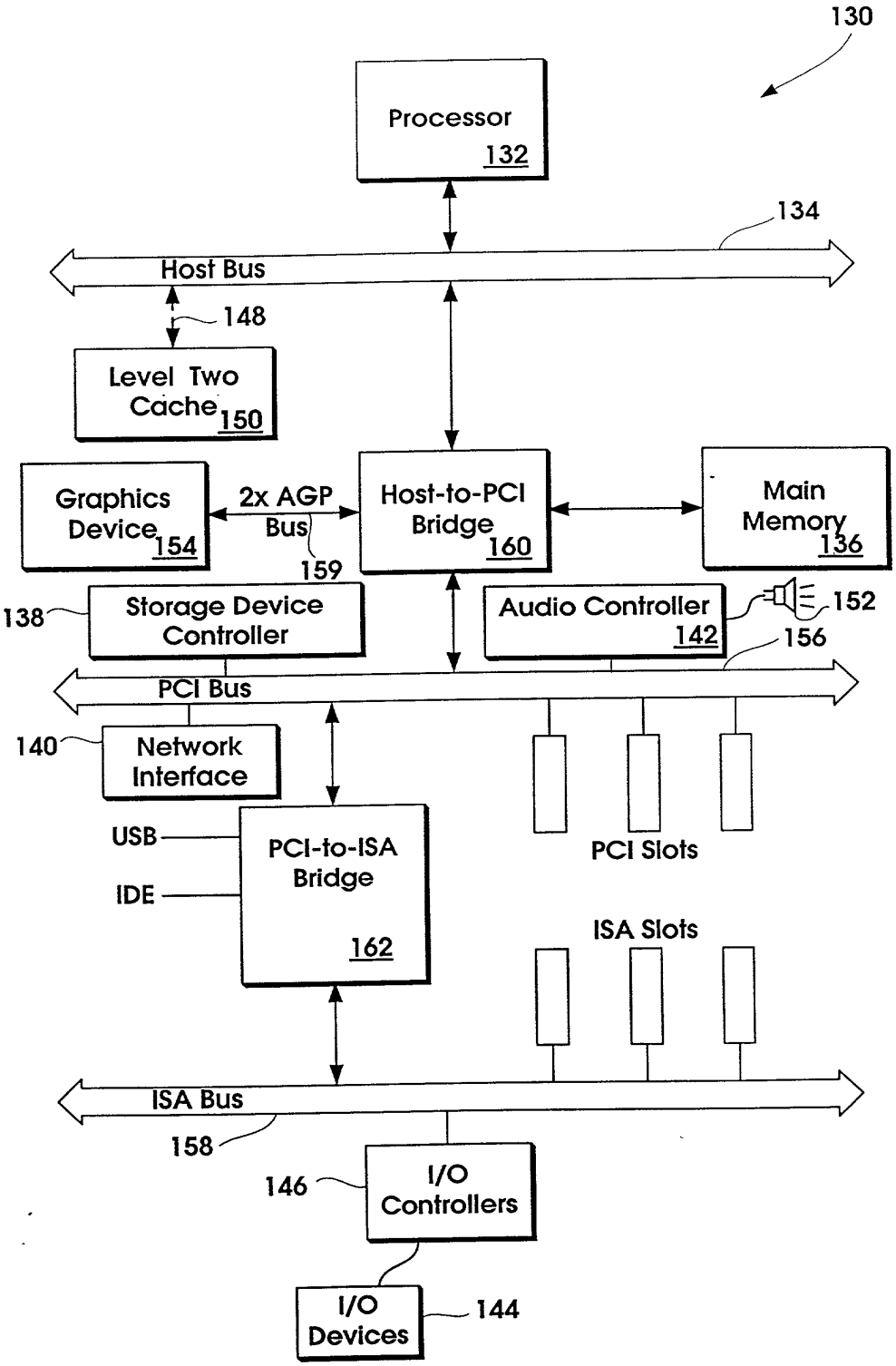

130
Processor
132
134
Host Bus
148
Level Two
Cache
150
Graphics
Device
154
2x AGP
Bus
159
Host-to-PCI
Bridge
160
Main
Memory
136
138
Storage Device
Controller
Audio Controller
142
152
156
PCI Bus
140
Network
Interface
USB
IDE
PCI-to-ISA
Bridge
162
PCI Slots
ISA Slots
ISA Bus
158
146
I/O
Controllers
I/O
Devices
144

FIG. 1

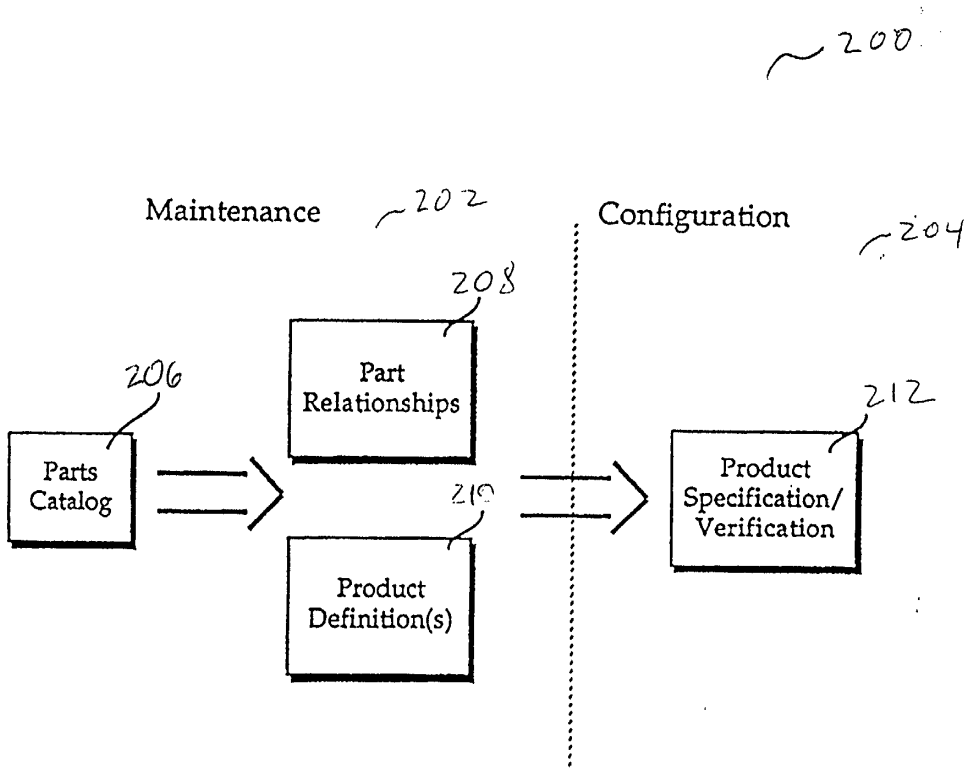


FIG. 2

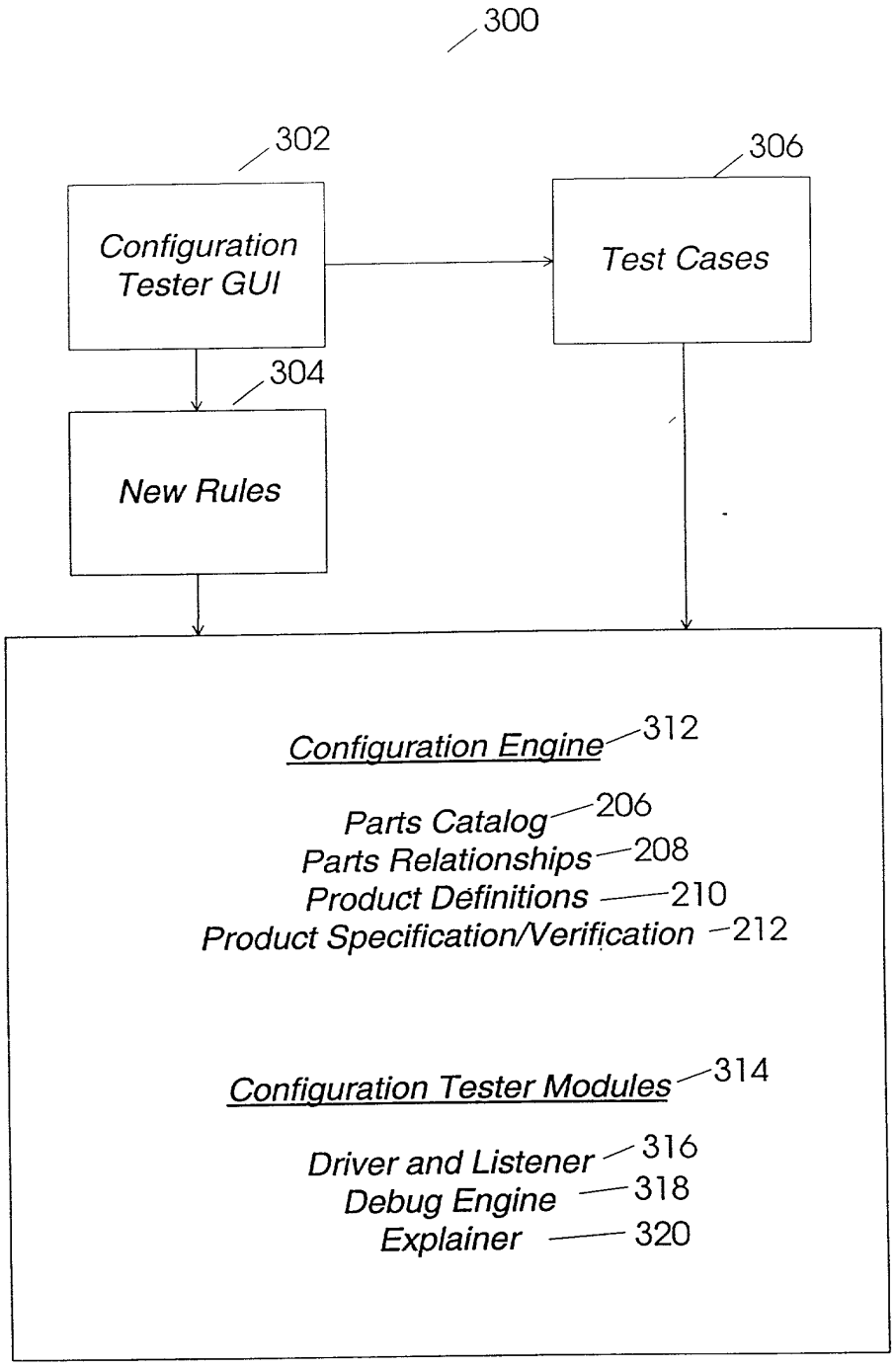

300
302
306
Configuration Tester GUI
Test Cases
304
New Rules
Configuration Engine
312
Parts Catalog
206
Parts Relationships
208
Product Definitions
210
Product Specification/Verification
212
Configuration Tester Modules
314
Driver and Listener
316
Debug Engine
318
Explainer
320

FIG. 3

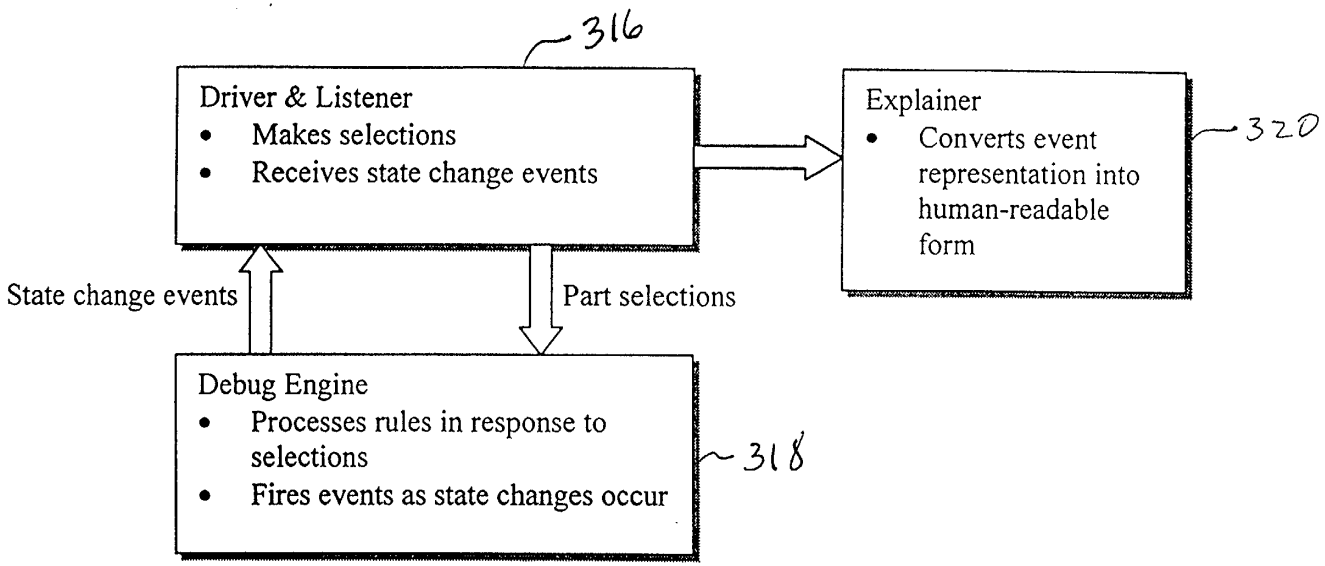

314
316
Driver & Listener
• Makes selections
• Receives state change events
Explainer
• Converts event representation into human-readable form
320
State change events
Part selections
Debug Engine
• Processes rules in response to selections
• Fires events as state changes occur
318

FIG. 3a

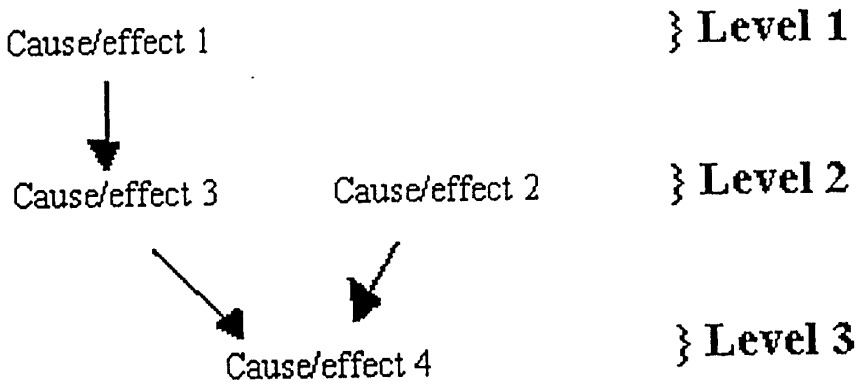

Cause/effect 1
Cause/effect 3
Cause/effect 2
Cause/effect 4
} Level 1
} Level 2
} Level 3

FIG. 3b

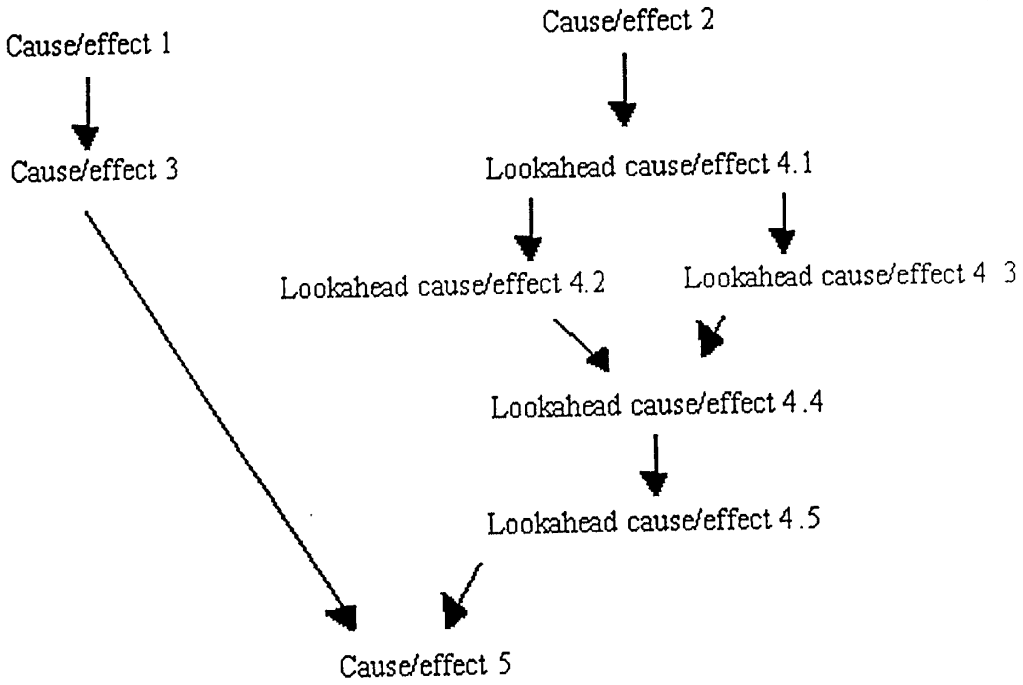

Cause/effect 1
Cause/effect 2
Cause/effect 3
Lookahead cause/effect 4.1
Lookahead cause/effect 4.2
Lookahead cause/effect 4 3
Lookahead cause/effect 4.4
Lookahead cause/effect 4.5
Cause/effect 5

Fig. 3c

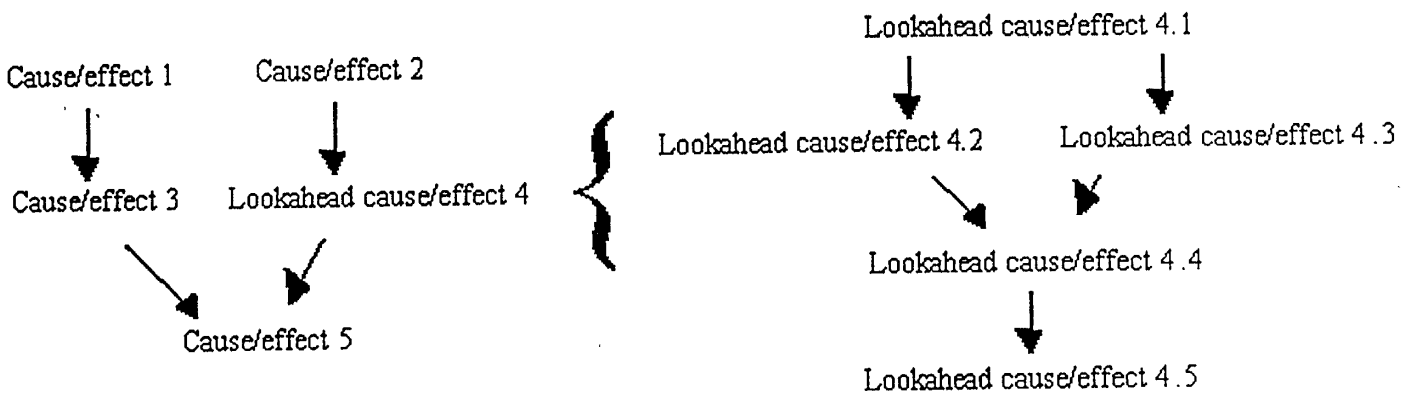


FIG. 3d

DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

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

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of subject matter (process, machine, manufacture, or composition of matter, or an improvement thereof) which is claimed and for which a patent is sought by way of the application entitled

Rule Based Configuration Engine For A Database

which (check) ☒ is attached hereto.
☐ and is amended by the Preliminary Amendment attached hereto.
☐ was filed on _____ as Application Serial No.
☐ and was amended on ___ (if applicable).

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

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

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

Prior Foreign Application(s)			Priority Claimed	
Number	Country	Day/Month/Year Filed	Yes	No
N/A			☐	☐

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

Provisional Application Number	Filing Date
N/A	

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

Application Serial No.	Filing Date	Status (patented, pending, abandoned)
N/A		

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith:

Alan H. MacPherson (24,423); Brian D. Ogonowsky (31,988); David W. Heid (25,875); Norman R. Klivans (33,003); Edward C. Kwok (33,938); David E. Steuber (25,557); Michael Shenker (34,250); Stephen A. Terrile (32,946); Peter H. Kang (40,350); Ronald J. Meetin (29,089); Ken John Koestner (33,004); Omkar K. Suryadevara (36,320); David T. Millers (37,396); Michael P. Adams (34,763); Robert B. Morrill (43,817); James E. Parsons (34,691); Philip W. Woo (39,880); Emily Haliday (38,903); Tom Hunter (38,498); Michael J. Halbert (40,633); Gary J. Edwards (41,008); Daniel P. Stewart (41,332); John T. Winburn (26,822); Tom Chen (42,406); Fabio E. Marino (43,339); Don C. Lawrence (31,975); Marc R. Ascolese (42,268); Carmen C. Cook (42,433); David G. Dolezal (41,711); Roberta P. Saxon (43,087); Mary Jo Bertani (42,321); Dale R. Cook (42,434); Sam G. Campbell (42,381); Matthew J. Brigham (44,047); Hugh H. Matsubayashi (43,779); Patrick D. Benedicto (40,909); T.J. Singh (39,535); Shireen Irani Bacon (40,494); Rory G. Bens (44,028); George Wolken, Jr. (30,441); John A. Odozynski (28,769); Cameron K. Kerrigan (44,826); Paul E. Lewkowicz (44,870); Theodore P. Lopez (44,881); Mayankkumar M. Dixit (44,064); Eric Stephenson (38,321); Christopher Allenby (45,906); David C. Hsia (46,235); Mark J. Rozman (42,117); Margaret M. Kelton (44,182); Do Te Kim (46,231); Alex Chen (45,591); Monique M. Heyninck (44,763); Gregory J. Michelson (44,940); Jonathan Geld (44,702); Emmanuel Rivera (45,760); Jason FarHadian (42,523); Matthew J. Spark (43,453); and Kent B. Chambers (38,839).

Please address all correspondence and telephone calls to:

Mary Jo Bertani
Attorney for Applicant(s)
SKJERVEN, MORRILL, MacPHERSON, FRANKLIN & FRIEL LLP
25 Metro Drive, Suite 700
San Jose, California 95110-1349

Telephone: 408-453-9200
Facsimile: 408-453-7979

I declare that all statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and all statements made herein are made with the knowledge that whoever, in any matter within the jurisdiction of the Patent and Trademark Office, knowingly and willfully falsifies, conceals, or covers up by any trick, scheme, or device a material fact, or makes any false, fictitious or fraudulent statements or representations, or makes or uses any false writing or document knowing the same to contain any false, fictitious or fraudulent statement or entry, shall be subject to the penalties including fine or imprisonment or both as set forth under 18 U.S.C. 1001, and that violations of this paragraph may jeopardize the validity of the application or this document, or the validity or enforceability of any patent, trademark registration, or certificate resulting therefrom.

Full name of sole (or first joint) inventor: Gilpin, Kevin E.

Inventor's Signature: _____ Date: 1/30/2001

Residence: Austin, Texas

Post Office Address: 4520 Highland Terrace
Austin, Texas 78731

Citizenship: United States

Full name of second inventor: Stein, Adam R.

Inventor's Signature: _____ Date: 11/29/2001

Residence: Mountain View, California

Post Office Address: 1310 Villa Street
Mountain View, California 94041

Citizenship: United States

PATENT NUMBER

ISSUE CLASSIFICATION: Class | Subclass

U.S. **UTILITY** Patent Application

O.I.P.E.	PATENT DATE
SCANNED ____ Q.A. ____	

APPLICATION NO.	CONT/PRIOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
09/773101		~~709~~ 702	45	~~2121~~ 2121	Stroke, Wilbert

APPLICANTS: Kevin Gilpin
Adam Stein

TITLE: Rule based configuration engine for a database

BEST AVAILABLE COPY

PTO-2040 12/99

ISSUING CLASSIFICATION

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

☐ TERMINAL DISCLAIMER	DRAWINGS			CLAIMS ALLOWED	
	Sheets Drwg.	Figs. Drwg.	Print Fig.	Total Claims	Print Claim for O.G.
☐ The term of this patent subsequent to _____ (date) has been disclaimed.				NOTICE OF ALLOWANCE MAILED	
	_____ (Assistant Examiner)		_____ (Date)		
☐ The term of this patent shall not extend beyond the expiration date of U.S Patent. No. _____				ISSUE FEE	
				Amount Due	Date Paid
	_____ (Primary Examiner)		_____ (Date)		
☐ The terminal ____months of this patent have been disclaimed.				ISSUE BATCH NUMBER	
	_____ (Legal Instruments Examiner)		_____ (Date)		

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

Form PTO-436A (Rev. 6/99)

FILED WITH: ☐ DISK (CRF) ☐ FICHE ☐ CD-ROM
(Attached in pocket on right inside flap)

(FACE)

SEARCHED

Class	Sub.	Date	Exmr.
706	045	13DEC03	JE
↓	046	↓	↓
709	317		
703	019		

SEARCH NOTES
(INCLUDING SEARCH STRATEGY)

	Date	Exmr.
EAST	13DEC03	JE
IEEE	↓	↓
ACM		
Internet		
(see Attached Sheets)		

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.

(RIGHT OUTSIDE)

ISSUE SLIP STAPLE AREA (for additional cross references)

POSITION	INITIALS	ID NO.	DATE
	Lth		02-05-02
FEE DETERMINATION			
O.I.P.E. CLASSIFIER		21	3/3/01
FORMALITY REVIEW	MM	76920	03-08-01
RESPONSE FORMALITY REVIEW	SS	593	06-25-01

INDEX OF CLAIMS

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

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

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

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

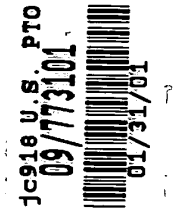
If more than 150 claims or 10 actions staple additional sheet here

(LEFT INSIDE)

02-01-01 A

SKJERVEN MORRILL MACPHERSON LLP

Docket No.: M-7822 US



January 31, 2001

Box Patent Application
Commissioner For Patents
Washington, D. C. 20231

Enclosed herewith for filing is a patent application, as follows:

Inventor(s): Gilpin, Kevin E.; Stein, Adam R.

Title: Rule Based Configuration Engine For A Database

X	Return Receipt Postcard
X	This Transmittal Letter (in duplicate)
21	page(s) Specification (not including claims)
11	page(s) Claims
1	page Abstract
7	Sheet(s) of Drawings
3	page(s) Declaration For Patent Application and Power of Attorney
1	page NonPublication Request
1	page(s) Recordation Form Cover Sheet (in duplicate)
2	page(s) Assignment

CLAIMS AS FILED

For	Number Filed			Number Extra		Rate			Basic Fee
								$	710.00
Total Claims	76	-20	=	56	x	$ 18.00	=	$	1,008.00
Independent Claims	8	-3	=	5	x	$80	=	$	400.00
☐ Fee of _____ for the first filing of one or more multiple dependent claims per application								$	

Please make the following charges to Deposit Account 19-2386:

☒ Total fee for filing the patent application in the amount of $ 2,118.00
☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account 19-2386.

EXPRESS MAIL LABEL NO:
EL764880543US

Respectfully submitted,

Mary Jo Bertani

Mary Jo Bertani
Attorney for Applicant(s)
Reg. No. 42,321

25 Metro Drive • Suite 700 • San Jose, CA 95110 • Phone (408) 453-9200 • Fax (408) 453-7979
San Francisco, CA • Austin, TX • Newport Beach, CA

719578 v1

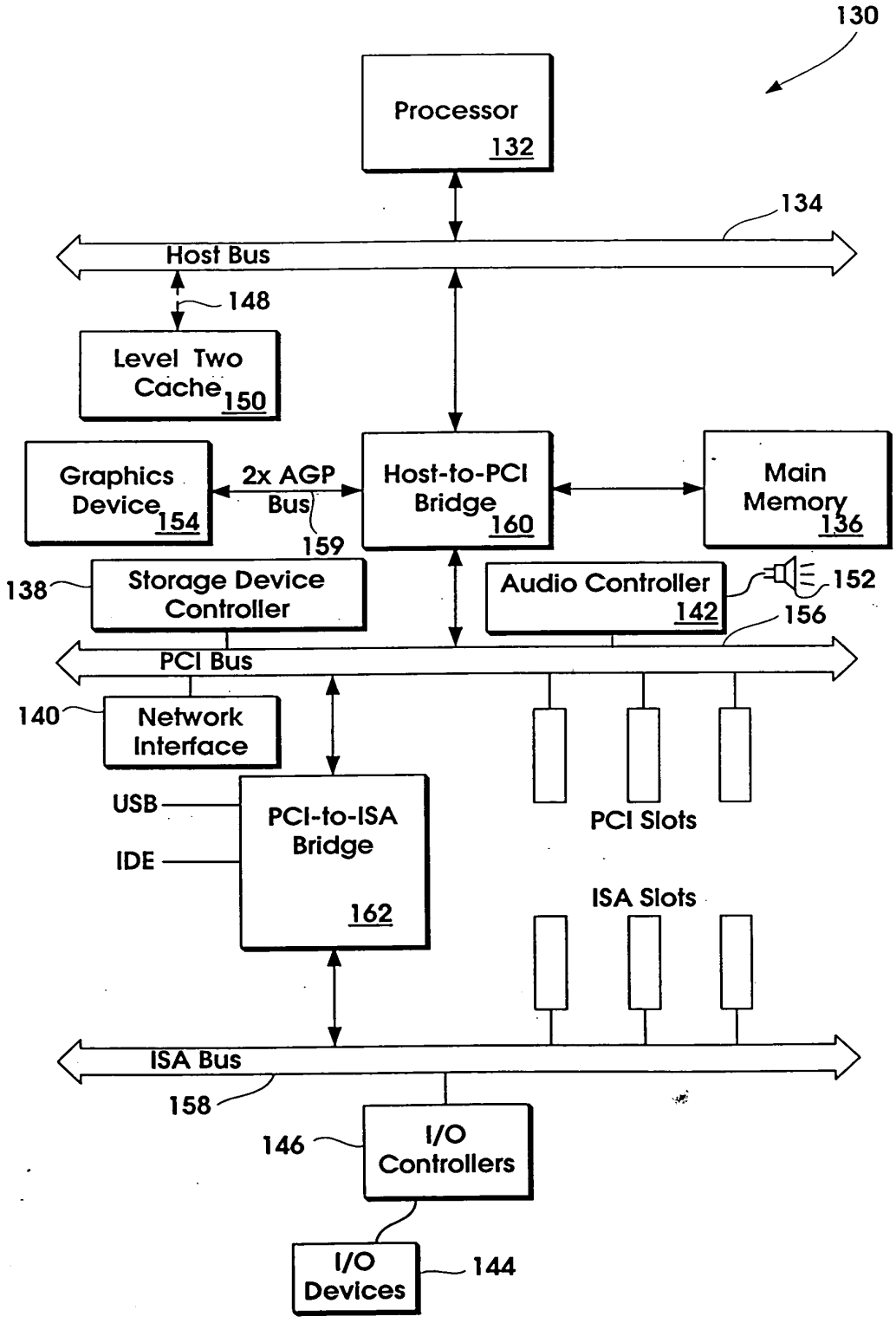

130
Processor
132
134
Host Bus
148
Level Two
Cache
150
Graphics
Device
154
2x AGP
Bus
159
Host-to-PCI
Bridge
160
Main
Memory
136
138
Storage Device
Controller
Audio Controller
142
152
156
PCI Bus
140
Network
Interface
USB
IDE
PCI-to-ISA
Bridge
162
PCI Slots
ISA Slots
ISA Bus
158
146
I/O
Controllers
I/O
Devices
144

FIG. 1

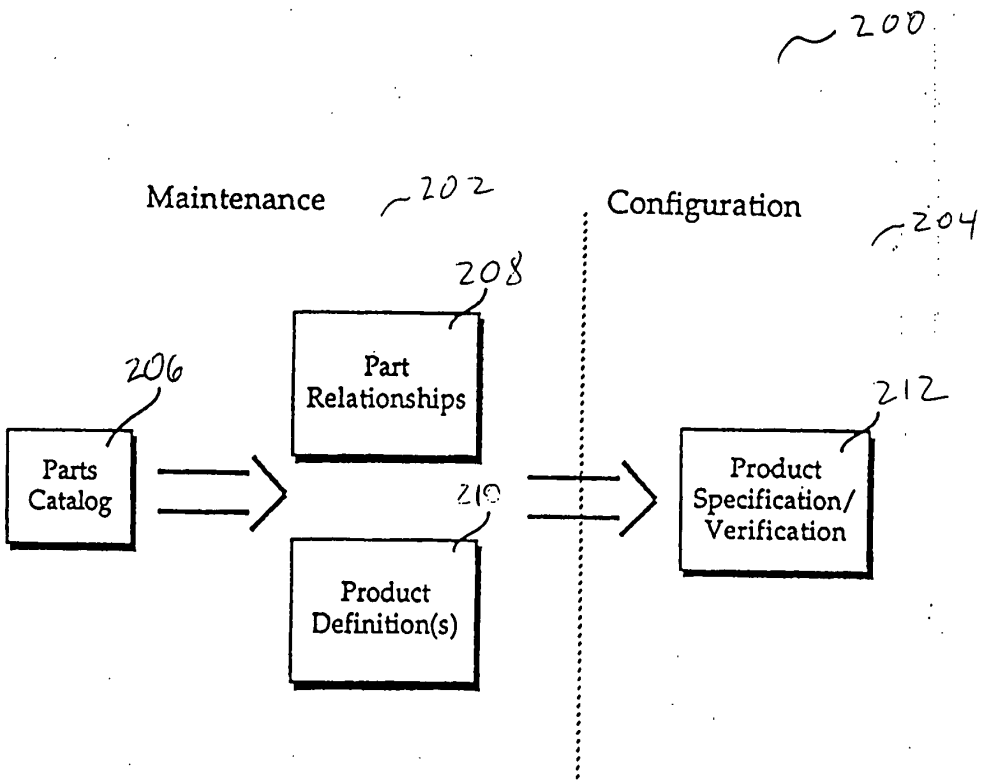

200
Maintenance
202
Configuration
204
208
206
Part Relationships
212
Parts Catalog
210
Product Specification/ Verification
Product Definition(s)

FIG. 2

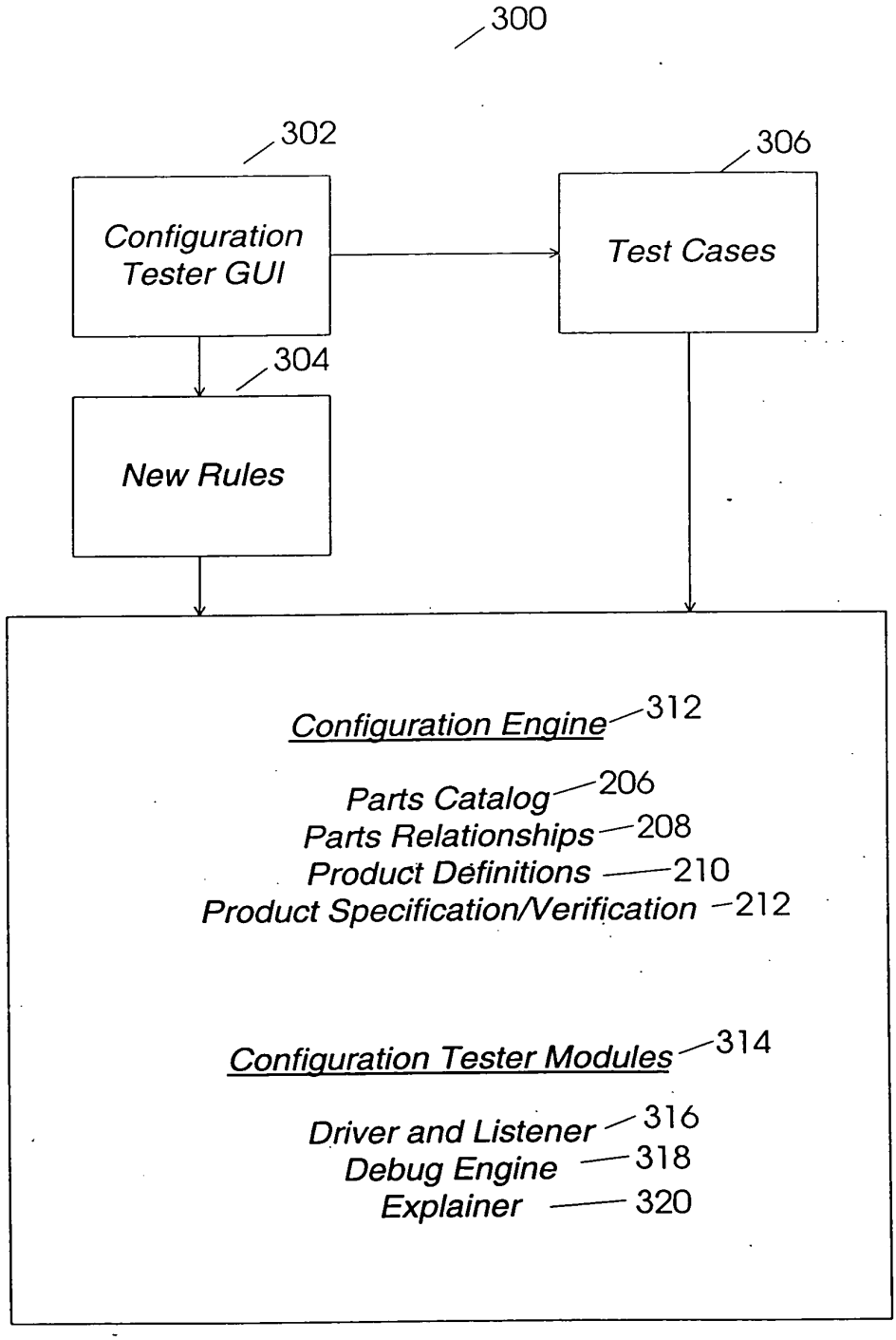

300
302
Configuration Tester GUI
306
Test Cases
304
New Rules
Configuration Engine
312
Parts Catalog
206
Parts Relationships
208
Product Definitions
210
Product Specification/Verification
212
Configuration Tester Modules
314
Driver and Listener
316
Debug Engine
318
Explainer
320

FIG. 3

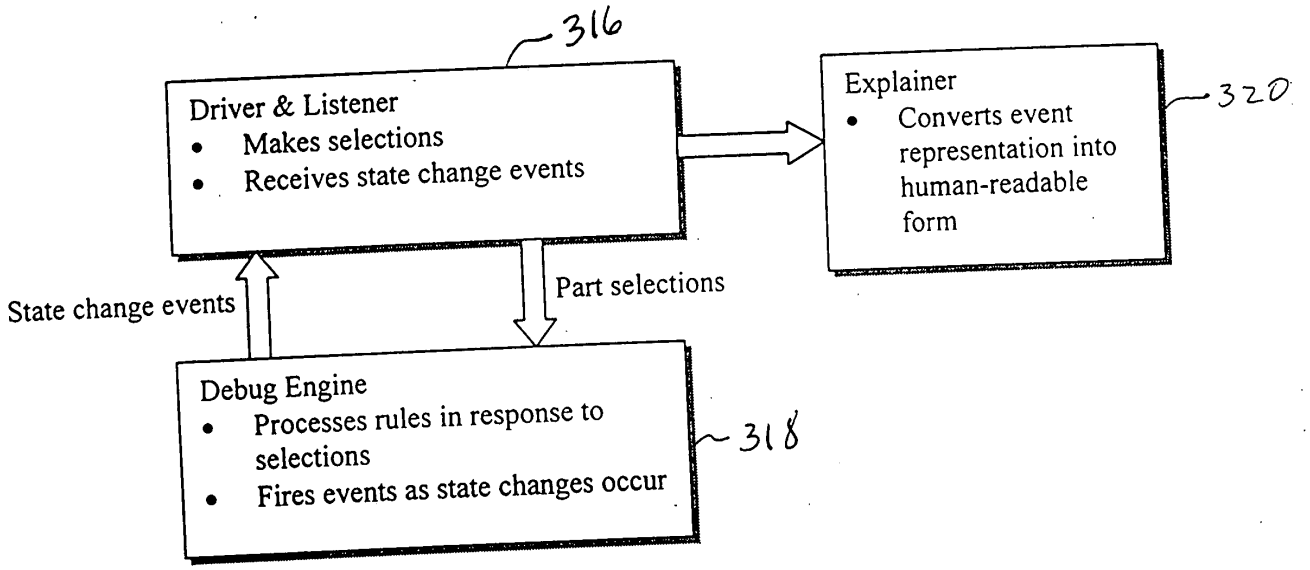


FIG. 3a

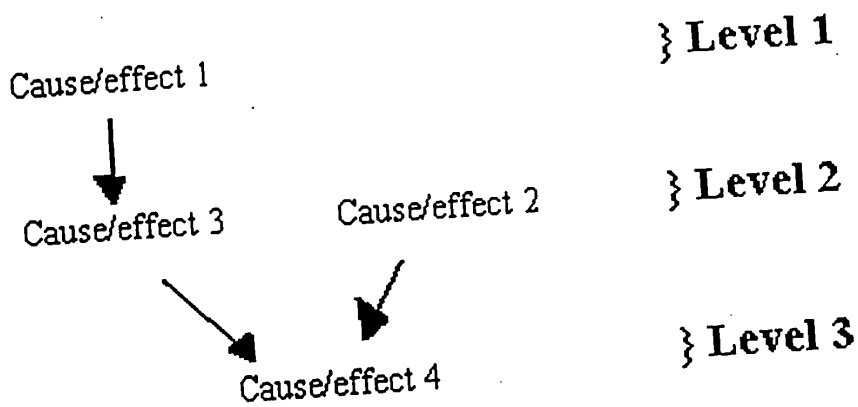

Cause/effect 1
Cause/effect 3
Cause/effect 2
Cause/effect 4
} Level 1
} Level 2
} Level 3

Fig. 3b

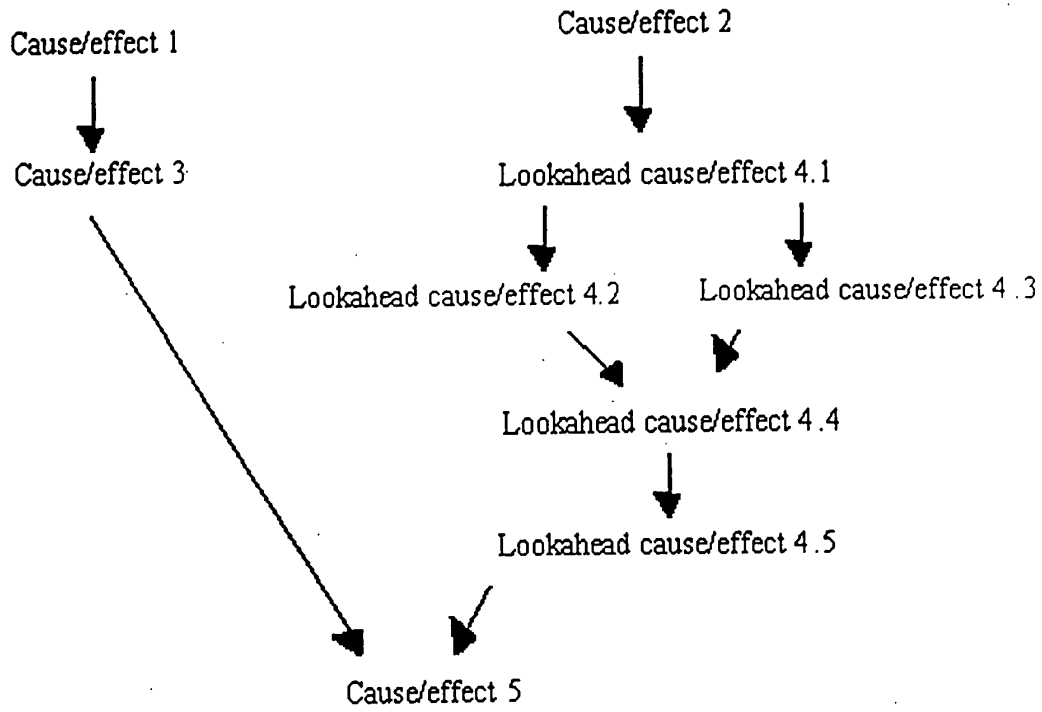


FIG. 3c

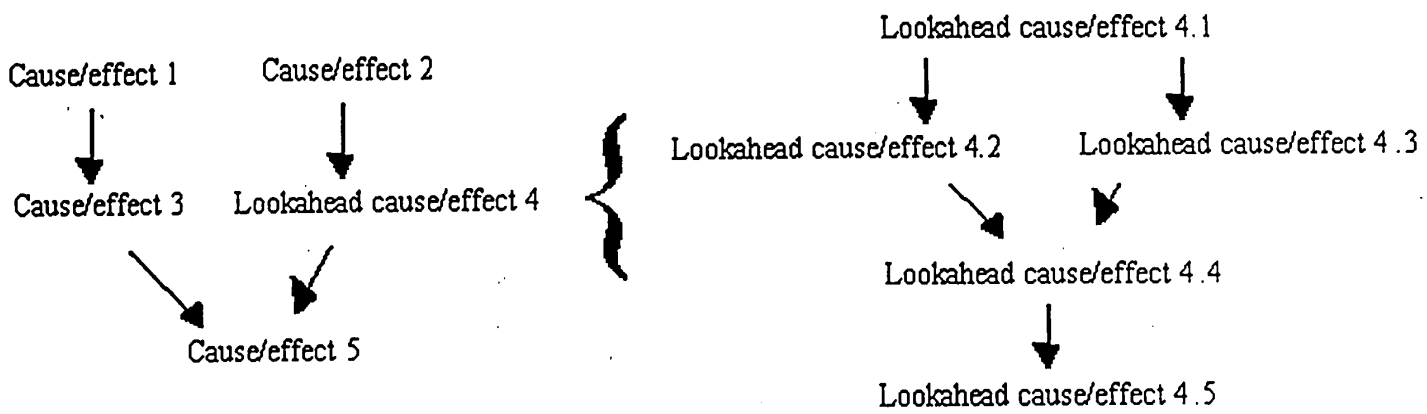

326
324
Cause/effect 1
Cause/effect 2
Cause/effect 3
Lookahead cause/effect 4
Cause/effect 5
Lookahead cause/effect 4.1
Lookahead cause/effect 4.2
Lookahead cause/effect 4 .3
Lookahead cause/effect 4 .4
Lookahead cause/effect 4 .5

FIG. 3d

"Express Mail" mailing label number:

EL764880543US

RULE BASED CONFIGURATION ENGINE FOR A DATABASE

Kevin E. Gilpin
Adam R. Stein

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates generally to computerized configuring systems. More specifically, this invention relates to a system and method for testing the compatibility of parts included in a configuration.

Description of the Related Art

Many products are comprised of individual parts or components. Currently, configuring systems, also referred to as configuration engines, are available that allow a user to configure a product by interactively selecting components from among various groups based on availability and compatibility of features and options for the product. One known system is described in U.S. Patent No. 5,825,651, entitled "Method and Apparatus For Maintaining and Configuring Systems," issued October 20, 1998, (hereinafter the "'651 patent"), which is assigned to the same assignee as the present invention, and is hereby incorporated by reference.

In one embodiment of a configuration system disclosed in the '651 patent, a framework for defining a product line includes a set of related components that are selected from a parts catalog. A product might consist of several hundred individual parts that are organized into part groups and are available on one or more of a number of products. A product is modeled by describing which parts and part groups are available in that product and which choices must be made from within the part groups, and then by writing additional rules that describe part-to-part relationships which are not modeled by the product structure. A compiler converts the product

structure and the rules into four rule types: includes (parts that are included by default), excludes, removes, and requires choice (a choice among a group of parts that must be made to achieve a valid configuration). Parts may also be classified as optional which signifies that they can be optionally included in the configuration.

After compilation, there may be several hundred, several thousand, or even more of these rules. When the system loads the model, all parts and products should initially be in a "selectable" state, which means that the client or user is allowed to choose them. When the client selects a part, that part is put in the "selected" state. Parts that are included by the selected parts enter the "included" state, and parts that are excluded by the selected parts enter the "excluded" state. Parts that were previously included but are removed by a "removes" rule are in the "deleted" state. Each part must always be in exactly one state. Parts that are selected by a user or are included are referred to as "selected". Parts that are excluded or deleted are referred to as "not selectable".

As product models grow in size and complexity, configuration errors may occur when a rule or series of rules is not properly defined and produces an undesired effect, such as the exclusion of a part that should be selectable. Configuration errors may also occur when a series of improperly defined rules causes a part to be in more than one state at the same time, such as "included" and "excluded", or "selected" and "deleted".

For large models, such errors may be difficult to find due to the large number of rules in the model, the unexpected effects of some configuration operations, and the complex interactions between rules. It is therefore desirable to have an automated testing tool to locate and analyze configuration errors, so that the rules may be corrected.

SUMMARY

The invention provides in one embodiment the ability to test rules in a rule-based system for configuring a product. A configuration system defines the

components of a product using elements contained in a parts catalog and rules that define relationships between the components of a product. The user provides test cases that select at least one part to include in the product configuration, and the configuration tester processes the rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

In one embodiment, the invention provides a method of testing a product configuration in a system for generating product configurations that include a variety of component parts. The configuration system includes one or more rules that define a relationship between at least two parts. The method includes entering a test case that selects at least one part to include in the product configuration. The system then processes the rule to determine whether part selected in the test case conflicts with parts that are already included in the product configuration, that is, whether the rule conflicts with other rules.

To test new rules, one embodiment of the invention initializes the configuration system with a part state and inputs at least one part selection as specified in the test case. A component referred to as a "listener" detects state change events that result in the system being in the initialized part state. Another component of the invention generates a cause that explains the part state in terms of the state change event, and generates a new part state for each part associated with the cause. The invention then determines the cause or causes that explain the new part states in terms of the state change event.

One feature of an embodiment of the invention generates a cause tree wherein the root of the cause tree is the initial part state, and "leaves" of the tree are the user's selections of parts.

Another component of an embodiment of the invention is an "explainer" which generates an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections. The explanation(s) are based on selection of a part, execution of a rule, a part being in two

states at the same time, a requires choice rule that cannot be satisfied, or on a look ahead process. To provide an explanation of how the system arrived at a particular part state, the invention sorts the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

In another embodiment, the invention is distributed as an article of manufacture, namely a computer usable medium having computer readable program code embodied therein for testing a product configuration in a system for generating product configurations. The system includes at least one rule defining a relationship between at least two parts, and the product configuration includes a plurality of parts.

The computer readable program code is configured to cause a computer to allow a user to enter a test case, wherein the test case selects at least one part to include in the product configuration. The program code then causes a computer to process the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration. This is accomplished by the computer readable program code causing a computer to initialize the system with a part state, to input the part selection to the system; and to listen to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

The program code then causes a computer system to determine the cause or causes that explain the new part states in terms of the state change event.

One feature of the program code causes a computer to generate a cause tree wherein the root of the cause tree is the initial part state, and "leaves" of the tree are the user's selections of parts.

Another component of the program code causes a computer to execute a component referred to as an "explainer" which generates an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections. The explanation(s) are based on selection of a part,

execution of a rule, a part being in two states at the same time, a requires choice rule that cannot be satisfied, or on a look ahead process. To provide an explanation of how the system arrived at a particular part state, the invention sorts the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

The foregoing has outlined rather broadly the objects, features, and technical advantages of the present invention so that the detailed description of the invention that follows may be better understood.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system with which the present invention may be utilized.

Figure 2 is a block diagram of an embodiment of a maintenance and configuration system with which the present invention may be utilized.

Figure 3 is a block diagram of a maintenance and configuration tester system according to an embodiment of the present invention.

Figure 3a is a block diagram of configuration tester modules included in an embodiment of the present invention.

Figure 3b is a diagram of an example of a cause/effect tree.

Figure 3c is a diagram of an example of a lookahead subtree embedded within a cause/effect tree.

Figure 3d is a diagram of an example of a lookahead subtree collapsed to a single node in the cause/effect tree.

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

DETAILED DESCRIPTION

A method and apparatus for testing a system for maintaining and configuring products is described. The present invention can be implemented on a general purpose computer system 130 such as illustrated in Fig. 1. Computer system 130 may be one of many workstations or servers connected to a network such as a local area network (LAN), a wide area network (WAN), or a global information network such as the Internet through network interface 140.

CPU 132 can be constructed from one or more microprocessors and/or integrated circuits. Main memory 136 stores programs and data that CPU 132 may access. When computer system 130 starts up, an operating system program is loaded into main memory 136. The operating system manages the resources of computer system 130, such as CPU 132, audio controller 142, storage device controller 138, network interface 140, I/O controllers 146, and host bus 134. The operating system reads one or more configuration files to determine the hardware and software resources connected to computer system 130.

During operation, main memory 136 includes the operating system, configuration file, and one or more application programs with related program data. Application programs can run with program data as input, and output their results as program data in main memory 136 or to one or more mass storage devices through a memory controller (not shown) and storage device controller 138. CPU 132 executes one or more application programs, including one or more programs to establish a connection to a computer network through network interface 140. The application programs may be embodied in one executable module or may be a collection of routines that are executed as required. Operating systems commonly generate "windows", as well known in the art, to present information about or from an

application program, and/or to allow a user to interact with an application program. Each application program typically has its own window that is generated when the application program is executing. Each window may be minimized to an icon, maximized to fill the display, overlaid in front of other windows, and underlaid behind other windows.

Storage device controller 138 allows computer system 130 to retrieve and store data from mass storage devices such as magnetic disks (hard disks, diskettes), and optical disks (DVD and CD-ROM). The information from the DASD can be in many forms including application programs and program data. Data retrieved through storage device controller 138 is usually placed in main memory 136 where CPU 132 can process it.

One skilled in the art will recognize that the foregoing components and devices are used as examples for sake of conceptual clarity and that various configuration modifications are common. For example, audio controller 142 is connected to PCI bus 156 in Fig. 1a, but may be connected to the ISA bus 138 or reside on the motherboard (not shown) in alternative embodiments. As further example, although computer system 130 is shown to contain only a single main CPU 132 and a single system bus 134, those skilled in the art will appreciate that the present invention may be practiced using a computer system that has multiple CPUs 132 and/or multiple busses 134. In addition, the interfaces that are used in the preferred embodiment may include separate, fully programmed microprocessors that are used to off-load computationally intensive processing from CPU 132, or may include input/output (I/O) adapters to perform similar functions. Further, PCI bus 156 is used as an exemplar of any input-output devices attached to any I/O bus; AGP bus 159 is used as an exemplar of any graphics bus; graphics device 154 is used as an exemplar of any graphics controller; and host-to-PCI bridge 160 and PCI-to-ISA bridge 162 are used as exemplars of any type of bridge. Consequently, as used herein the specific exemplars set forth in Fig. 1 are intended to be representative of their more general classes. In general, use of any specific exemplar herein is also intended to be representative of its class and the non-inclusion of such specific devices in the foregoing list should not be taken as indicating that limitation is desired.

The invention detects and analyzes configuration errors in a system for configuring products such as described in the '651 patent. A brief description of the '651 patent is provided in the following paragraphs as background for understanding the present invention.

Brief Description Of The '651 Patent

Referring to Fig. 2, one embodiment of configuration engine 200 disclosed in the '651 patent is shown. Configuration engine 200 is rule-based, and includes maintenance environment 202 and configuration environment 204. Maintenance environment 202 includes zero or more individual parts, or components, in parts catalog 206. Part relationships 208 defines relationships between a first set of parts and a second set of parts so that when all of the members of the first set of parts are selected, the relationship between the two sets is enforced on the parts in the second set. A set of parts can include multiple parts. The incorporation of parts in a set can be arbitrary. That is, a multi-part set can contain parts that are otherwise unrelated. For example, for the purpose of configuring an automobile, a set can contain parts such as an engine, sun roof, and a color. These parts seem to be unrelated, but they can be combined into a part relationship 208 for purposes of forming a relationship using an embodiment of configuration engine 200.

In one embodiment, there are four kinds of part relationships 208 between parts: requires choice, includes, excluded, and removes. An included part is included automatically. A part is excluded from the configuration when its inclusion would result in an invalid configuration. A part may be removed when another part is added. Thus, when a first part exists in the configuration and a second part is added, the first part is removed from the configuration if there is a conflict. The requires choice relationship is used to allow a set of choices to be made from a group of parts. The number of parts chosen is limited to a valid bounds specification. The relations that are created between parts within a product are enforced only on that particular product. However, if some part relationships 208 are to be enforced on all products within a product line, then the relations are generated once and enforced for all products.

One or more product definitions 210 are generated by a population of component parts. Using configuration engine 200, a user can configure a product given product definitions 210 and part relationships 208 associated with product definitions 210. Configuration environment 204 accepts a configuration user's input and processes it in product specification/verification 212 to verify the product configuration, and to update the specification based on the user's input, or to notify the user that the input is invalid based on product definitions 210 and user selections.

A graphical user interface (GUI) is used to allow the user to interactively generate product definitions 210. GUI operations such as drag, drop, and selection operations can be used to specify product definitions 210.

Relationships associated with items contained in product definitions 210 are evaluated when user input is received. Configuration engine 200 determines which relationships are active and inactive given the user input. A relationship is active when all the items in a product's product definition 210 are selected. A relationship is inactive until all of the parts in a product's product definition 210 are selected.

Configuration engine 200 is used to configure a product using a definition created by the maintenance environment 202. Configuration environment 204 ensures that the current configuration state is always valid. The user can select and unselect parts in any order. When user input is received, product specification/verification 212 validates the input based on the current state of the configuration. In addition, the product specification/verification 212 identifies selections that could cause a valid configuration to become invalid. Product specification/verification 212 removes these selections from the set of possible selections so that the user does not make an invalid selection.

Configuration engine 204 evaluates the current state of a configuration based on product definitions 210, part relationships 208, and state information. After receipt of input from a user, product specification/verification 212 evaluates relationships in both the forward and backward direction. Forward and backward evaluations can result in the addition or deletion of elements from the product configuration.

During configuration, information is stored in tables and vectors. Configuration engine 200 represents elements in a configuration (e.g., product, part, and group) as bits in a bit vector. Thus, for example, a vector includes a number of bits is equal to the total number of elements. An element's bit can be set or reset to specify the state of the element in the current configuration. For example, a user vector can be used that specifies for each element whether the element has been selected by the user during the configuration. In addition, excluded and removed vectors identify whether an element is excluded or removed (respectively) from a configuration. Vectors can be used to identify whether an element 1) has been selected (by the user or the configuration system), 2) is selectable, and 3) not selectable.

Tables contain element relationships. A table is used to represent the includes, excludes, removes, and requires choice relationships, for example. Each table has a left-hand side and a right-hand side that corresponds to the left-hand and right-hand sides of a relationship. In each case, the left-hand side is a bit vector that contains bits corresponding to elements. The includes, excludes and removes tables contain a bit vector in the right-hand side that represents configuration elements. The right-hand side of the requires choice table is a pointer that points to an entry in a group table. The group table entry is a bit vector that identifies the elements that are contained in the group from which a choice is to be made. The right-hand side of a requires choice table entry further includes minimum and maximum designations. Minimum and maximum values identify the minimum and maximum number of group members that are to be selected to satisfy a requires group relationship.

The bit vector implementation of relationships and internal runtime state allows for fast and efficient computation of relationship-based configuration. A comparison of bits can be performed in one machine instruction in most cases.

There are many ways that errors can be introduced into a configuration, however, the effects of these errors can be categorized in 2 groups:

1) A part is put in a state which was not intended by the user (state error), or

2) A part is put in more than one state at the same time (exception error).

Errors may be caused by a single rule, or by a chain of rules. Complex errors are often caused by a "look ahead" process included in product specification/verification 212 that test-selects each product (if more than one product is selectable) to make sure that it is in fact selectable. The look-ahead process helps insure that the state of a product is not reported as selectable when selection of that product would lead to a rule conflict. The look-ahead process also determines the sets of parts that are excluded or deleted by every selectable product.

Further errors may arise with requires choice rules, which typically require that between some minimum (*min*) and maximum (*max*) number of choices must be made from a set of parts. For example, there is always an implicit requires choice rule that specifies that at least exactly one (min/max = 1/1) part must be selected for a product. Requires choice rules are complex to evaluate because they may cause many kinds of inferences. In general, there is no way to determine whether a selectable part is actually selectable without selecting it and checking to see whether it causes a conflict. In order to ensure that each selectable part is not going to cause such a conflict, configuration engine 200 would have to select a selectable part after each user selection, which is too computationally expensive. The following examples of each type of error illustrate the problem.

State Errors

The simplest types of state errors are caused when a rule has been accidentally omitted or written. For example, the user may select PartA and PartB, and then note that 'PartC' is excluded rather than selectable. In the simplest case, this may be due to the following rule in the model:

PartA Excludes PartC

Or, there may be a rule:

PartA Requires Choice (min/max = 1/1) { PartB, PartC }

Here, selecting PartA implies that either PartB or PartC must be selected. Selecting PartB causes configuration engine 200 to infer that PartC must be Excluded. Alternatively, multiple rules may cause a state change, for example:

PartA Includes PartX
PartX Excludes PartC

Here, selecting PartA causes PartX to be included, which then causes PartC to be excluded.

State errors can also be caused by the look-ahead process. Suppose the following rules are written:

ProductA Excludes PartA
ProductB Includes PartB
ProductB, PartB Excludes PartA
ProductC RequiresChoice (min/max 1/1) PartA, PartC
ProductC Includes PartC

Even if the user has not made any selections, PartA will be excluded by the look ahead process for each of products A, B, and C. Detecting state changes that are caused by the look-ahead process is particularly difficult because for each product there may be a different rule chain or exception error that causes the state error.

Exception Errors

Sometimes, rules may be inadvertently written that cause a conflicting state exception. The simplest case can be summed up by the rules:

PartA includes PartB

PartB excludes PartA

If PartA is selected, then PartB will be Included. But then the second rule causes PartA to be excluded. This conflicting state cannot be reconciled, and an exception is raised.

Most exception conditions are more complex than this one. For example, selecting a part that is in a requires choice rule may cause the requires choice rule to be unsatisfiable as follows:

PartA requires choice (min/max=1/1) { PartB, Part C }
PartB includes PartC

In the preceding rules, if PartA is selected, selecting PartB will cause an exception error because the requires choice rule will not be satisfiable.

Configuration Testing

Fig. 3 shows an embodiment of the present invention for configuration tester system 300 that includes several components for detecting and analyzing configuration errors. One component is configuration tester graphical user interface (CTGUI) 302 that enables users to enter new rules 304 and define test cases 306 that describe the expected behavior of their models to test the configuration. New rules 304 are input to parts relationships 308 and product definitions 310 in configuration engine 312. Test cases 306 describe one or more sets of selections that should be made, and sets of parts and their expected states based on new rules 304, as well as rules previously included in parts relationships 308 and product definitions 310. For example, test cases 306 may describe the selection of a product and several parts. It may then ensure that some other set of parts is excluded, and a third set is included. An example of a test case in test cases 306 is:

Select ProductA
Select PartA
Ensure that (PartB, PartC) are excluded

Ensure that (PartD) is included

Once test cases 306 are written, configuration tester modules 314 run each test case 306 and verify that the tested parts are in the right state. If a test fails, configuration tester modules 314 determine why a part is in a certain state and explain the state as described below. The database of pre-existing rules can then be modified to correct errors found by configuration tester modules 314.

Configuration tester modules 314 include driver and listener module 316, debug engine 318, and explainer 320. Fig. 3a shows interrelationships of configuration tester modules 314 including types of data communicated between driver and listener 316, debug engine 318, and explainer 320, during operation. Driver and listener 316 selects parts from test cases 306 and sends the part selections to debug engine 318.

Debug engine 318 processes new rules 304 using the part selections, and sends state change events to driver and listener 316 as state changes result from the rules executing, exceptions occurring, and execution of the look ahead process. In the '651 patent, configuration engine 200 (Fig. 2) is optimized for very high performance. In one embodiment, configuration tester system 300 includes components of configuration engine 200 such as parts catalog 206, parts relationships 208, product definitions 210, and product specification/verification 210. Configuration tester system 300 can run in test mode or normal mode so that no performance penalties are imposed when operating configuration tester system 300 in normal mode. This is accomplished by executing all features and components of configuration tester system 300 from debug engine 318, which is only used in test mode.

The application program interface (API) to debug engine 318 includes program instructions to include new rules 304 with existing rules in parts relationships 208 and product definitions 210, and to run test cases 306 through product specification/verification 210. Debug engine 318 presents the same API as the normal mode of configuration engine 200 for selecting parts. CTGUI 302 is used to specify which test cases to run. Whenever a condition occurs that causes a part state

change, debug engine 318 detects this condition and transmits an appropriate notice to driver and listener 316 for the listener portion to handle and interpret the events.

Driver and listener 316 listens to the state change events and constructs a tree of the rule chains that are executing in debug engine 318 and resulting states. When a state error occurs, driver and listener 316 executes a driver to recreate the error condition for the part for which the state error occurred, along with all the part selections that caused the error to occur. The combination of the part and its state is represented by a part state.

In one embodiment, the part-state includes an identifier for the part, the state of the part, the selections which have been made (which are always a subset of the total user selections), and, optionally, the product for which lookahead is currently being run. For example, a part-state may represent:

Part A is included after selecting Part X and Part Y,
or
Part B is excluded with no selections during lookahead on Product X.

Each part-state also has a Cause, which is initially null. Configuration tester system 300 determines the Cause of the part state (a rule firing, an exception, a user selection, etc) and sets the Cause of the part-state.

Driver and listener 316 interfaces with debug engine 318. The driver portion of driver and listener 316 starts submitting the part selections that led to the error until a state change event occurs that recreates the error condition. The listener portion of driver and listener 316 is responsible for handling the state change events. It may happen as a result of any of the following:

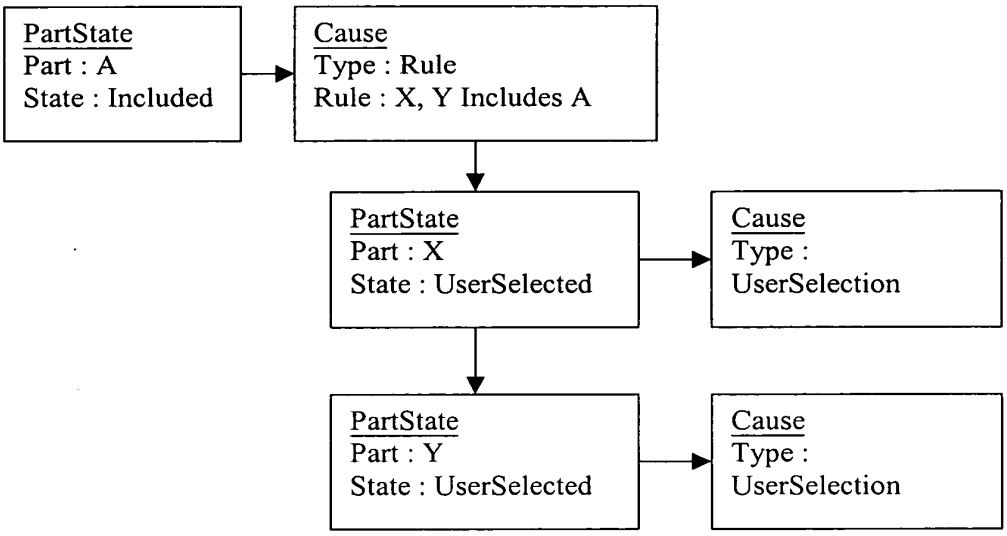
1) A user selection
2) A rule executing
3) A rule conflict (exception error)
4) Operation of the look ahead process

In each case, the driver generates a cause, which represents the event and the state change that resulted from it. Then, based on this new information, further analysis to explain the part state may be required to explain the error in accordance with the following summary:

Cause	Explanation Complete?	Next steps
User Selection	Yes	
Rule Executing	No	Determine why the rule executed
Conflicting State Exception (part is in 2 states at the same time)	No	Explain each of the conflicting states
Unsatisfiable requires choice exception	No	Determine why the requires choice rule executed, and explain the state of the parts that caused it to be unsatisfiable
Look ahead process	No	Explain the state of the part in each selectable product

Driver and listener 316, and debug engine 318, are recursive. The driver portion of driver and listener 316 is initialized with a single part state, along with a set of user selections. The user selections are specified in the test case. The driver inputs each user selection one by one, until the listener gets a state change event that explains the part state. Then the listener generates a cause that explains the part state in terms of the event. The listener also generates a new part state for each part associated with the cause. Then driver and listener 316 start over to find the causes that explain the new part states. Eventually, all part states can be explained in terms of a user selection. The explanation of the original part state is thus represented by a tree of causes and part states. The original part state is the root of the tree. The second level of the tree, i.e., the leaves, consist of the causes that caused the root part state. The next level is the part states that caused the causes, and so on.

For example, in one embodiment, suppose the task is: Explain why A is Included if X and Y are UserSelected. The tree might look like this:



Each PartState points to its Cause. If the Cause is a RuleCause, the Cause points to the parts that caused the rules to fire and their state is in turn explained with Cause objects.

Explainer 320 converts the tree into a format that readily allows the user to visualize the rules that are causing an erroneous result in the configured product. The root of the tree is the initial goal part state, and the leaves of the tree are the user's selections of parts. It is more intuitive to the user, however, to see the part selections as the root of the explanation, and then the chain of causes that follow from these selections. Accordingly, explainer 320 accepts the tree as input, and generates a description of the sequence of events by modeling the logical operation of configuration engine 312, not the literal sequence of actions. This is because converting the tree requires more than post-order traversal, which only provides a trace of the state of configuration engine 312. Logically, configuration engine 312 operates in a series of cause-and-effects iterations. In each iteration, configuration engine 312 first determines which rules should execute, and then applies the results of those rules to the current state of the configuration. The process then repeats until the

internal state of the configuration is no longer changing with each iteration. At this point, equilibrium is reached, and configuration engine 312 is ready to once again receive another selection of a part as input.

Explainer 320 determines the stem for each cause in a given iteration from part states in previous iterations, and determines the cause for each part state in the same iteration. This provides a mechanism for grouping and sorting the tree by iteration. In the simple case, the iteration number of a given part state is determined by measuring the longest distance between a part state and a leaf cause. For any given node in the cause/effect tree, the distances between that node and all the leaves of the tree that connect to that node can be counted. The maximum of this set of values is the maximum depth of the node, which is also the iteration number for that cause/effect. Fig. 3b shows an example of a cause/effect tree where the maximum depth of cause/effect 4 is two (2) (level 3 minus level 1).

Consider, for example, the following set of rules:

1) A includes X
2) B excludes Y
3) A,C,X require Y,Z

And the following sequence of events:

1) User picks A
2) Rule 1 brings in part X
3) User picks B
4) Rule 2 excludes Y
5) User picks C
6) Rule 3 includes Z

There are several things to notice in this example. First, the order of user selections is irrelevant with regard to the logical operation of configuration engine 312 is concerned. Also, the order of execution of rules 1 and 2 is irrelevant. These details are abstracted away when the sequence of events is broken into logical rounds:

Round 1: User selects A, B, C
Round 2: Rule 1 includes X
Rule 2 excludes Y
Round 3: Rule 3 includes Z

The latter description eases understanding the logical flow of configuration engine 312, and better highlights the dependencies between user actions and rules. This is especially true in situations involving more complex series of rules. For the preceding example, the latter representation makes it immediately clear that the activations of rules 1 and 2 are not causally linked events, whereas the first representation leaves open the possibility that rule 2 executes as a consequence of rule 1.

Complications Caused By Look Ahead

In the look ahead process, configuration engine 312 makes a series of selections to determine what would happen if the user chose particular parts. Many rules can execute within a particular look ahead scenario, but eventually all of these rule executions are retracted, and the results of the look ahead process are applied to the current product being configured. Therefore, an entire look ahead event happens within an individual round of configuration engine 312 activity, even though the look ahead event itself may contain many rounds of executing rules. The recursive aspect of the causes and part states tree is taken into account to invert the causes and part states tree with explainer 320. Essentially, explainer 320 regards look ahead events as branches within the main tree, and collapses them down to single nodes when calculating the proper round in which to place a given cause or part state. An example of what happens during look ahead is: given two products, P1 and P2, and the rules 'P1 excludes A', 'P2 excludes A', Lookahead internally selects each selectable product in turn, and determines whether there are any parts which are excluded by all products. In this example, A would be excluded by lookahead. To the explainer, this can be summarized as 'A is excluded by Lookahead', but within each product, the rules provide a further cause.

Figs. 3c and 3d show how lookahead nodes are collapsed to a single node of the main cause/effect tree. Specifically, Fig. 3c shows lookahead nodes 4.1 though 4.5 expanded within the main cause/effect tree 322, while Fig. 3d shows lookahead subtree 324 collapsed into lookahead cause/effect 4 in main cause/effect tree 326.

In one embodiment, Explainer 320 is designed in an object-oriented fashion that allows key elements of the process to be overridden to provide specialized behavior. For example, some configuration models are generated automatically from known product data descriptions or other sources. Explainer 320 can be overridden to trace explanations all the way back to these original rule sources. Explainer 320 can also be overridden to integrate data from historical logs or databases, as well as data input by the user.

The present invention has been described in the context of a fully functional computer system, however those skilled in the art will appreciate that the present invention is capable of being distributed as a program product in a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of signal bearing media include: recordable type media such as floppy disks and CD-ROM, transmission type media such as digital and analog communications links, as well as media storage and distribution systems developed in the future.

Additionally, the foregoing detailed description has set forth various embodiments of the present invention via the use of block diagrams, flowcharts, and examples. It will be understood by those within the art that each block diagram component, flowchart step, and operations and/or components illustrated by the use of examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or any combination thereof. In one embodiment, the present invention may be implemented via Application Specific Integrated Circuits (ASICs). However, those skilled in the art will recognize that the embodiments disclosed herein, in whole or in part, can be equivalently implemented in standard integrated circuits, as a computer program running on a computer, as firmware, or as virtually any combination thereof and that designing the circuitry and/or writing the

code for the software or firmware would be well within the skill of one of ordinary skill in the art in light of this disclosure.

While the invention has been described with respect to the embodiments and variations set forth above, these embodiments and variations are illustrative and the invention is not to be considered limited in scope to these embodiments and variations. Accordingly, various other embodiments and modifications and improvements not described herein may be within the spirit and scope of the present invention, as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method of testing a product configuration in a system for generating product configurations, the system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the method comprising:

entering a test case, wherein the test case selects at least one part to include in the product configuration; and

processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

2. The method, as set forth in claim 1, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

initializing the system with a part state;

inputting the at least one part selection; and

listening to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

3. The method, as set forth in claim 2, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

generating a cause that explains the part state in terms of the state change event.

4. The method, as set forth in claim 3, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

generating a new part state for each part associated with the cause.

5. The method, as set forth in claim 4, wherein processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration, further includes:

determining the causes that explain the new part states in terms of the state change event.

6. The method, as set forth in claim 5, further comprising:

generating a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

7. The method, as set forth in claim 6, further comprising:

generating an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

8. The method, as set forth in claim 7, wherein the explanation is based on selection of a part.

9. The method, as set forth in claim 7, wherein the explanation is based on execution of a rule.

10. The method, as set forth in claim 7, wherein the explanation is based on a part being in two states at the same time.

11. The method, as set forth in claim 7, wherein the explanation is based on a requires choice rule that cannot be satisfied.

12. The method, as set forth in claim 7, wherein the explanation is based on a look ahead process.

13. The method, as set forth in claim 7, further comprising:

sorting the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

14. An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for testing a product configuration in a system for generating product configurations, the system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the computer readable program code including:

computer readable program code configured to cause a computer to allow a user to enter a test case, wherein the test case selects at least one part to include in the product configuration; and

computer readable program code configured to cause a computer to process the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

15. The article of manufacture, as set forth in claim 14, further including:

computer readable program code configured to cause a computer to initialize the system with a part state;

computer readable program code configured to cause a computer to input the at least one part selection; and

computer readable program code configured to cause a computer to listen to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

16. The article of manufacture, as set forth in claim 15, further including:

computer readable program code configured to cause a computer to generate a cause that explains the part state in terms of the state change event.

17. The article of manufacture, as set forth in claim 16, further including:

computer readable program code configured to cause a computer to generate a new part state for each part associated with the cause.

18. The article of manufacture, as set forth in claim 17, further including:

computer readable program code configured to cause a computer to determine the causes that explain the new part states in terms of the state change event.

19. The article of manufacture, as set forth in claim 18, further comprising:
computer readable program code configured to cause a computer to generate a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

20. The article of manufacture, as set forth in claim 19, further comprising:
computer readable program code configured to cause a computer to generate an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

21. The article of manufacture, as set forth in claim 20, wherein the explanation is based on selection of a part.

22. The article of manufacture, as set forth in claim 20, wherein the explanation is based on execution of a rule.

23. The article of manufacture, as set forth in claim 20, wherein the explanation is based on a part being in two states at the same time.

24. The article of manufacture, as set forth in claim 20, wherein the explanation is based on a requires a choice rule that cannot be satisfied.

25. The article of manufacture, as set forth in claim 20, wherein the explanation is based on a look ahead process.

26. The article of manufacture, as set forth in claim 20, further comprising:
computer readable program code configured to cause a computer to sort the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

27. An apparatus for testing a product configuration generated by a product configuration system, comprising:

at least one rule defining a relationship between at least two parts in the product configuration;

a test case pertaining to at least one part to include in the product configuration; and

a processor coupled to receive the at least one rule and the test case, wherein the processor is operable to determine whether the at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to the at least one rule.

28. The apparatus, as set forth in claim 27, wherein the processor is further operable to:

initialize the configuration system with a part state;

to input the at least one part selection;

to listen to state change events in the system; and

to detect when a state change event occurs that results in the configuration system being in the initialized part state.

29. The apparatus, as set forth in claim 28, wherein the processor is further operable to:

generate a cause that explains the part state in terms of the state change event.

30. The apparatus, as set forth in claim 29, wherein the processor is further operable to:

generate a new part state for each part associated with the cause.

31. The apparatus, as set forth in claim 30, wherein the processor is further operable to:

generate a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

32. The apparatus, as set forth in claim 30, wherein the processor is further operable to:

generate an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

33. The apparatus, as set forth in claim 32, wherein the explanation is based on execution of a rule.

34. The apparatus, as set forth in claim 32, wherein the explanation is based on a part being in two states at the same time.

35. The apparatus, as set forth in claim 32, wherein the explanation is based on a requires a choice rule that cannot be satisfied.

36. The apparatus, as set forth in claim 32, wherein the explanation is based on a look ahead process.

37. The apparatus, as set forth in claim 30, wherein the processor is further operable to:

sort the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

38. A configuration system comprising:

a modified database, wherein the modified database is based on the results of testing a product selection using a test case, wherein the test case includes at least one part selection, and the expected state of the at least one selected part.

39. The configuration system of claim 38, wherein the test case further includes the product selection.

40. The configuration system of claim 38, further comprising:

at least one vector, wherein said vector comprises a bit field, further wherein the bit field comprises bits that represent elements in a configuration.

41. The configuration system of claim 40, wherein the number of bits in the bit field is equal to the total number of elements and an element's bit can be set or reset to specify the state of the element in the configuration.

42. The configuration system of claim 40, wherein the vector specifies whether an element has been selected by the user during the configuration.

43. The configuration system of claim 40, wherein excluded vectors identify whether an element is excluded from a configuration.

44. The configuration system of claim 40, wherein removed vectors identify whether an element is removed from a configuration.

45. The configuration system of claim 40, wherein the vector identifies whether an element is selectable.

46. A database comprising:

at least one table, wherein said table represents relationships between elements in a configuration; and

at lease one modified rule, wherein the rule is modified based on the results of testing a product selection.

47. The database of claim 46, wherein said table represents "includes" relationships between elements in a configuration.

48. The database of claim 46, wherein said table represents "excludes" relationships between elements in a configuration.

49. The database of claim 46, wherein said table represents "removes" relationships between elements in a configuration.

50. The database of claim 46, wherein said table represents "requires choice" relationships between elements in a configuration.

51. The database of claim 50, wherein the representation of "requires choice" relationships includes a pointer to a group table that includes a bit vector that identifies the elements that are contained in the group from which a choice is to be made.

52. The database of claim 50, wherein the representation of "requires choice" relationships includes minimum and maximum designations to identify the minimum and maximum number of group members that are to be selected to satisfy the "requires choice" relationship.

53. The database of claim 46, wherein said table includes a left-hand side and a right-hand side.

54. The database of claim 53, wherein the left-hand side includes a bit vector that contains bits corresponding to elements.

55. The database of claim 53, wherein the right-hand side includes one or more bit vectors that represent configuration elements.

56. A test case for testing a product configuration generated by a product configuration system, comprising:

a product selection;

at least one part selection; and

an expected state of the selected part based on one or more rules.

57. A method for identifying an invalid configuration generated by a product configuration system, comprising:

selecting a product;

selecting at least one part; and

generating a part state of the selected part based on one or more rules.

58. The method as set forth in claim 57, further comprising:

generating a cause that explains the part state in terms of a state change event.

59. The method as set forth in claim 57, further comprising:

detecting an error in the part state.

60. The method as set forth in claim 57, further comprising:

detecting when the at least one part is put in more than one state at the same time.

61. The method as set forth in claim 57, further comprising:

determining whether the product is selectable.

62. The method as set forth in claim 57, further comprising:

detecting when the at least one part is put in more than one state at the same time.

63. The method as set forth in claim 57, further comprising:

reporting the state of the product as not selectable when selection of the product would conflict with the rule.

64. The method as set forth in claim 57, further comprising:

determining sets of parts that are excluded or deleted based on the product.

65. The method as set forth in claim 57, further comprising:

detecting when a state change event occurs that results in the system being in the initialized part state.

66. The method as set forth in claim 65, further comprising:
generating a cause that explains the part state in terms of the state change event.

67. The method, as set forth in claim 66, further comprising:
generating a new part state for each part associated with the cause.

68. The method, as set forth in claim 67, further comprising:
generating a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

69. The method, as set forth in claim 68, further comprising:
generating an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

70. An apparatus for testing a product configuration generated by a product configuration system, comprising:
means for defining a relationship between at least two parts in the product configuration;
means for defining a test case for at least one part to include in the product configuration; and
means for determining whether the at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to at least one rule.

71. The apparatus, as set forth in claim 70, further comprising:
means for initializing the configuration system with a part state;
means for detecting a state change event in the system; and
means for detecting when a state change event occurs that results in the configuration system being in the initialized part state.

72. The apparatus, as set forth in claim 71, further comprising:

means for generating a cause that explains the part state in terms of the state change event.

73. The apparatus, as set forth in claim 72, further comprising:
means for generating a new part state for each part associated with the cause.

74. The apparatus, as set forth in claim 73, further comprising:
means for generating a cause tree, wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

75. The apparatus, as set forth in claim 73, further comprising:
means for generating an explanation of the part state, wherein the part selections are the root of the explanation and the causes follow from the part selections.

76. The apparatus, as set forth in claim 70, further comprising:
means for modifying the at least one rule when the test case conflicts with the plurality of parts previously included in the product configuration.

RULE BASED CONFIGURATION ENGINE FOR A DATABASE

Kevin E. Gilpin
Adam R. Stein

Abstract of the Disclosure

The invention provides the ability to test rules in a rule-based system for configuring a product. The configuration system defines the components of a product using elements contained in a parts catalog and rules that define relationships between the components of a product. The user provides test cases that select at least one part to include in the product configuration, and the configuration tester processes the rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

Attorney Docket No.: M-7822 US

DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

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

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of subject matter (process, machine, manufacture, or composition of matter, or an improvement thereof) which is claimed and for which a patent is sought by way of the application entitled

Rule Based Configuration Engine For A Database

which (check) ☒ is attached hereto.
☐ and is amended by the Preliminary Amendment attached hereto.
☐ was filed on _____ as Application Serial No.
☐ and was amended on __ (if applicable).

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

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

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

Prior Foreign Application(s)			Priority Claimed	
Number	Country	Day/Month/Year Filed	Yes	No
N/A			☐	☐

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

Provisional Application Number	Filing Date
N/A	

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

Application Serial No.	Filing Date	Status (patented, pending, abandoned)
N/A		

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith:

Alan H. MacPherson (24,423); Brian D. Ogonowsky (31,988); David W. Heid (25,875); Norman R. Klivans (33,003); Edward C. Kwok (33,938); David E. Steuber (25,557); Michael Shenker (34,250); Stephen A. Terrile (32,946); Peter H. Kang (40,350); Ronald J. Meetin (29,089); Ken John Koestner (33,004); Omkar K. Suryadevara (36,320); David T. Millers (37,396); Michael P. Adams (34,763); Robert B. Morrill (43,817); James E. Parsons (34,691); Philip W. Woo (39,880); Emily Haliday (38,903); Tom Hunter (38,498); Michael J. Halbert (40,633); Gary J. Edwards (41,008); Daniel P. Stewart (41,332); John T. Winburn (26,822); Tom Chen (42,406); Fabio E. Marino (43,339); Don C. Lawrence (31,975); Marc R. Ascolese (42,268); Carmen C. Cook (42,433); David G. Dolezal (41,711); Roberta P. Saxon (43,087); Mary Jo Bertani (42,321); Dale R. Cook (42,434); Sam G. Campbell (42,381); Matthew J. Brigham (44,047); Hugh H. Matsubayashi (43,779); Patrick D. Benedicto (40,909); T.J. Singh (39,535); Shireen Irani Bacon (40,494); Rory G. Bens (44,028); George Wolken, Jr. (30,441); John A. Odozynski (28,769); Cameron K. Kerrigan (44,826); Paul E. Lewkowicz (44,870); Theodore P. Lopez (44,881); Mayankkumar M. Dixit (44,064); Eric Stephenson (38,321); Christopher Allenby (45,906); David C. Hsia (46,235); Mark J. Rozman (42,117); Margaret M. Kelton (44,182); Do Te Kim (46,231); Alex Chen (45,591); Monique M. Heyninck (44,763); Gregory J. Michelson (44,940); Jonathan Geld (44,702); Emmanuel Rivera (45,760); Jason FarHadian (42,523); Matthew J. Spark (43,453); and Kent B. Chambers (38,839).

Please address all correspondence and telephone calls to:

Mary Jo Bertani
Attorney for Applicant(s)
SKJERVEN, MORRILL, MacPHERSON, FRANKLIN & FRIEL LLP
25 Metro Drive, Suite 700
San Jose, California 95110-1349

Telephone: 408-453-9200
Facsimile: 408-453-7979

I declare that all statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and all statements made herein are made with the knowledge that whoever, in any matter within the jurisdiction of the Patent and Trademark Office, knowingly and willfully falsifies, conceals, or covers up by any trick, scheme, or device a material fact, or makes any false, fictitious or fraudulent statements or representations, or makes or uses any false writing or document knowing the same to contain any false, fictitious or fraudulent statement or entry, shall be subject to the penalties including fine or imprisonment or both as set forth under 18 U.S.C. 1001, and that violations of this paragraph may jeopardize the validity of the application or this document, or the validity or enforceability of any patent, trademark registration, or certificate resulting therefrom.

Full name of sole (or first joint) inventor: Gilpin, Kevin E.

Inventor's Signature: _____ Date: 1/30/2001

Residence: Austin, Texas

Post Office Address: 4520 Highland Terrace
Austin, Texas 78731

Citizenship: United States

Full name of second inventor: Stein, Adam R.

Inventor's Signature: _____ Date: 11/29/2001

Residence: Mountain View, California

Post Office Address: 1310 Villa Street
Mountain View, California 94041

Citizenship: United States

PATENT APPLICATION SERIAL NO. _____

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

02/06/2001 SSITHIB1 00000001 192386 09773101

01 FC:101 710.00 CH
02 FC:103 1008.00 CH
03 FC:102 400.00 CH

PTO-1556
(5/87)

*U.S. GPO: 1999-459-082/19144

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

Application or Docket Number: M-7822 US

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	76	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	76 minus 20=	* 56
INDEPENDENT CLAIMS	8 minus 3 =	* 5
MULTIPLE DEPENDENT CLAIM PRESENT		☐

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

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	1,008
X40=		OR	X80=	400
+135=		OR	+270=	
TOTAL		OR	TOTAL	2118

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

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

FORM PTO-875
(Rev. 8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-460-706/30103

EXPRESS MAIL LABEL
EL764880540US

#4
10/0/01

MODIFIED PTO/SB/35 (11-00)

REQUEST AND CERTIFICATION UNDER 35 U.S.C. 122(b)(2)(B)(i)	Inventors	Gilpin, Kevin E.; Stein, Adam R.
	Title	Rule Based Configuration Engine For A Database
	Atty Docket Number	M-7822 US

I hereby certify that the invention disclosed in the attached application **has not and will not be** the subject of an application filed in another country, or under a multilateral agreement, that requires publication at eighteen months after filing. I hereby request that the attached application not be published under 35 U.S.C. 122(b).

January 31, 2001
Date

Mary Jo Bertani
Attorney for Applicants
Reg. No.: 42,321

This request must be signed in compliance with 37 CFR 1.33(b) and submitted with the application **upon filing.**

Applicant may rescind this nonpublication request at any time. If applicant rescinds a request that an application not be published under 35 U.S.C. 122(b), the application will be scheduled for publication at eighteen months from the earliest claimed filing date for which a benefit is claimed.

If applicant subsequently files an application directed to the invention disclosed in the attached application in another country, or under a multilateral international agreement, that requires publication of applications eighteen months after filing, the applicant **must** notify the United States Patent and Trademark Office of such filing within forty-five (45) days after the date of the filing of such foreign or international application. **Failure to do so will result in abandonment of this application (35 U.S.C. 122(b)(2)(B)(iii)).**

7 CFR 1.213(a) provides for a request that an application not be published under 35 U.S.C. 122(b). Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. SEND TO: Commissioner for Patents, Washington, DC 20231.

#2

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US

CONFIRMATION NO. 5458

FORMALITIES LETTER

OC000000005843769

Mary Jo Bertani
SKJERVEN, MORRILL, MacPHERSON,
FRANKLIN & FRIEL LLP
25 Metro Drive, Suite 700
San Jose, CA 95110-1349

Date Mailed: 03/09/2001

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

This application has been accorded an Application Number and Filing Date. The application, however, is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given TWO MONTHS from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a)

The required item(s) identified below must be timely submitted to avoid abandonment:

- Substitute drawings in compliance with 37 CFR 1.84 because:
  - drawing sheets do not have the appropriate margin(s) (see 37 CFR 1.84(g)). Each sheet must include a top margin of at least 2.5 cm. (1 inch), a left side margin of at least 2.5 cm. (1 inch), a right side margin of at least 1.5 cm. (5/8 inch), and a bottom margin of at least 1.0 cm. (3/8 inch);

A copy of this notice MUST be returned with the reply.

Customer Service Center
Initial Patent Examination Division (703) 308-1202
PART 3 - OFFICE COPY

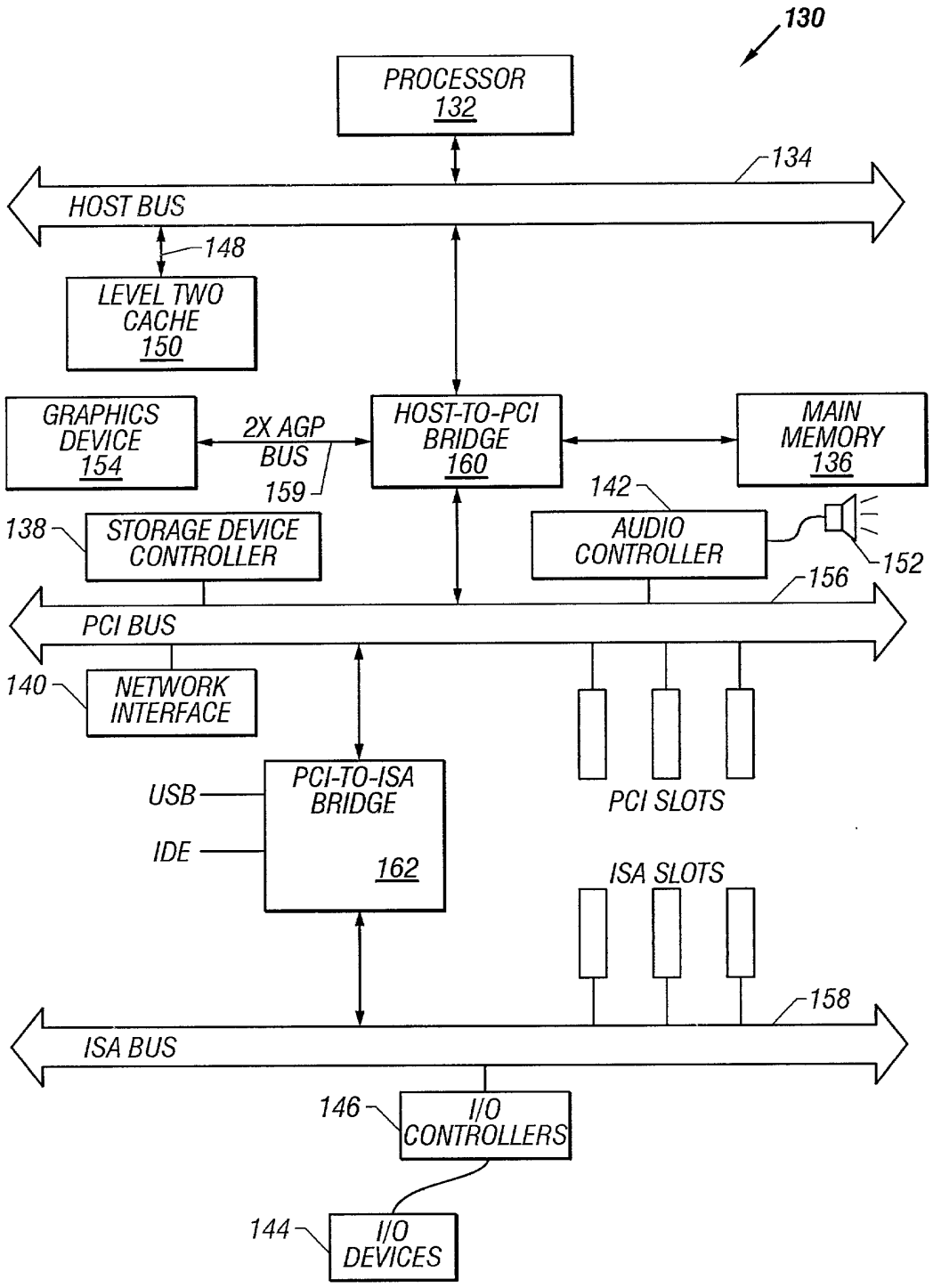

130
PROCESSOR
132
134
HOST BUS
148
LEVEL TWO
CACHE
150
GRAPHICS
DEVICE
154
2X AGP
BUS
159
HOST-TO-PCI
BRIDGE
160
MAIN
MEMORY
136
142
138
STORAGE DEVICE
CONTROLLER
AUDIO
CONTROLLER
152
156
PCI BUS
140
NETWORK
INTERFACE
USB
IDE
PCI-TO-ISA
BRIDGE
162
PCI SLOTS
ISA SLOTS
158
ISA BUS
146
I/O
CONTROLLERS
144
I/O
DEVICES

FIG. 1

2/6

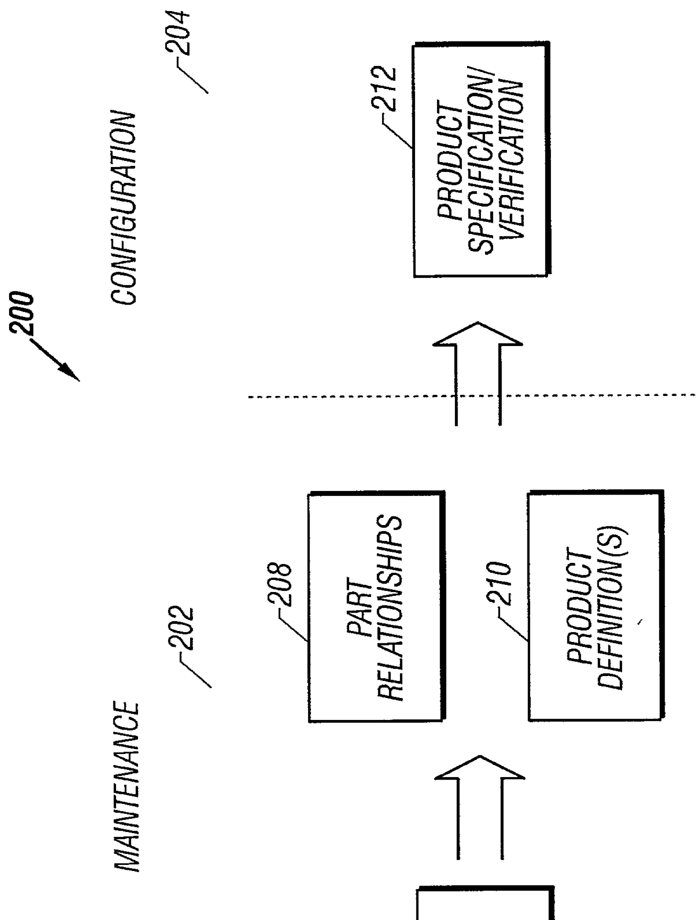

200
MAINTENANCE
202
CONFIGURATION
204
PARTS CATALOG
206
PART RELATIONSHIPS
208
PRODUCT DEFINITION(S)
210
PRODUCT SPECIFICATION/ VERIFICATION
212

FIG. 2

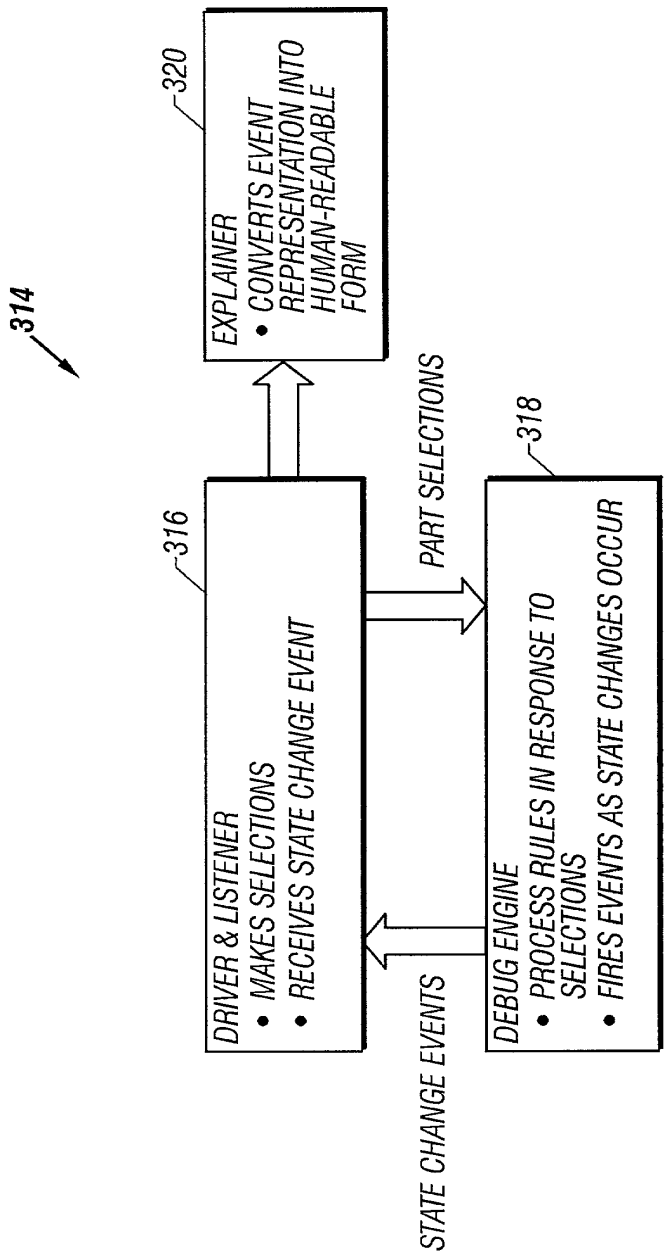

314
320
EXPLAINER
• CONVERTS EVENT REPRESENTATION INTO HUMAN-READABLE FORM
316
DRIVER & LISTENER
• MAKES SELECTIONS
• RECEIVES STATE CHANGE EVENT
PART SELECTIONS
318
DEBUG ENGINE
• PROCESS RULES IN RESPONSE TO SELECTIONS
• FIRES EVENTS AS STATE CHANGES OCCUR
STATE CHANGE EVENTS

FIG. 3A

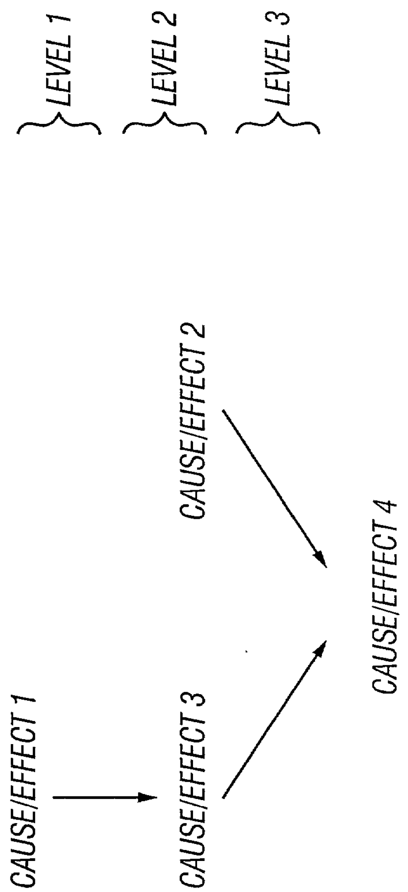

LEVEL 1
LEVEL 2
LEVEL 3
CAUSE/EFFECT 1
CAUSE/EFFECT 2
CAUSE/EFFECT 3
CAUSE/EFFECT 4

FIG. 3B

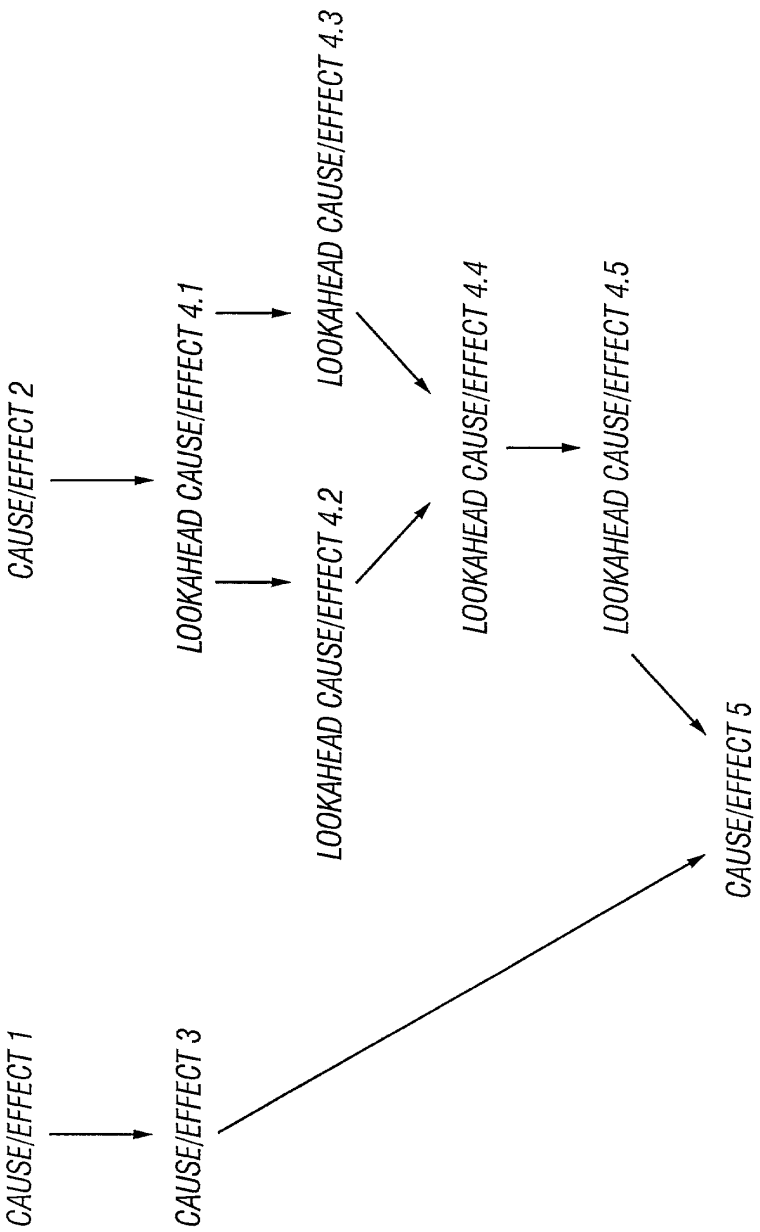

322
CAUSE/EFFECT 1
CAUSE/EFFECT 2
CAUSE/EFFECT 3
LOOKAHEAD CAUSE/EFFECT 4.1
LOOKAHEAD CAUSE/EFFECT 4.2
LOOKAHEAD CAUSE/EFFECT 4.3
LOOKAHEAD CAUSE/EFFECT 4.4
LOOKAHEAD CAUSE/EFFECT 4.5
CAUSE/EFFECT 5

FIG. 3C

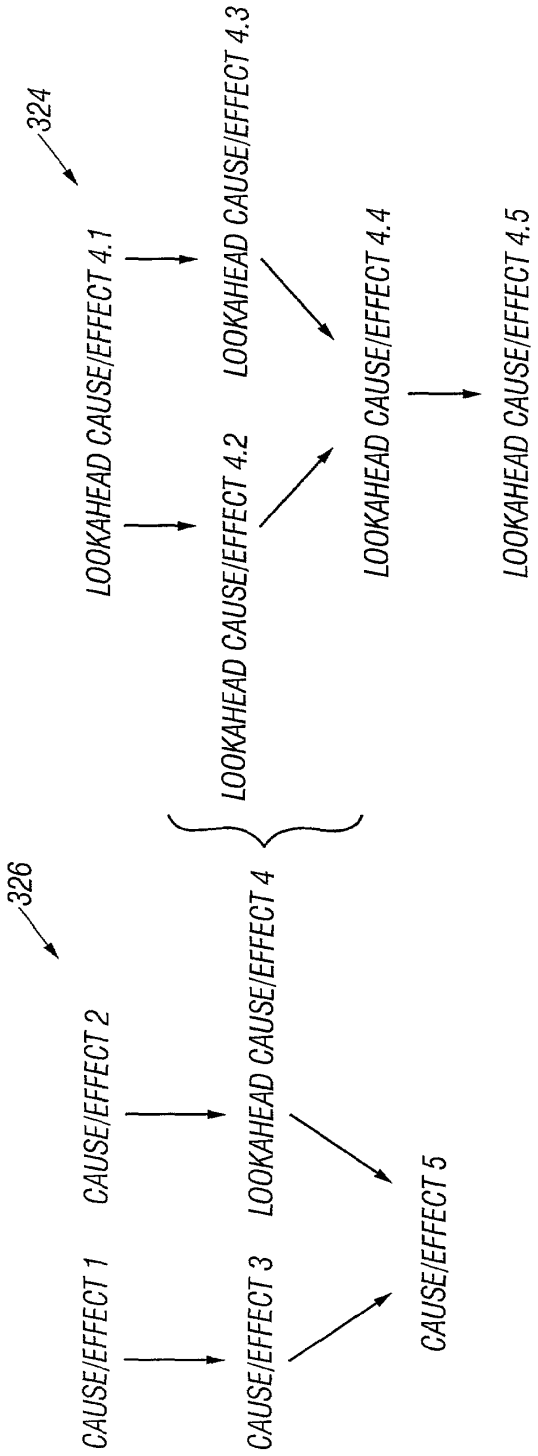
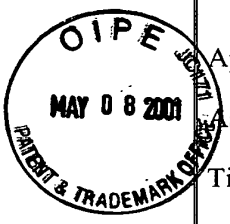

324
LOOKAHEAD CAUSE/EFFECT 4.1
LOOKAHEAD CAUSE/EFFECT 4.3
LOOKAHEAD CAUSE/EFFECT 4.2
LOOKAHEAD CAUSE/EFFECT 4.4
LOOKAHEAD CAUSE/EFFECT 4.5
326
CAUSE/EFFECT 2
LOOKAHEAD CAUSE/EFFECT 4
CAUSE/EFFECT 1
CAUSE/EFFECT 3
CAUSE/EFFECT 5

FIG. 3D

05-09-01 0300

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#3



Applicant(s):	Gilpin, Kevin E.; Stein, Adam R.		
Assignee:	Trilogy Development Group, Inc.		
Title:	Rule Based Configuration Engine For A Database		
Serial No.:	09/773,101	Filing Date:	January 31, 2001
Examiner:	Not Assigned	Group Art Unit:	2121
Docket No.:	M-7822 US		

San Jose, California
May 8, 2001

Attn: Official Draftsperson
COMMISSIONER FOR PATENTS
Washington, D. C. 20231

SUBMISSION OF FORMAL DRAWINGS IN RESPONSE TO NOTICE TO FILE CORRECTED APPLICATION PAPERS

Dear Sir:

Applicants submit six (6) sheets of formal drawings, consisting of Figures 1, 2, 3A, 3B, 3C, and 3D in the above-named application. This is being submitted in response to the Notice to File Corrected Application Papers mailed by the United States Patent and Trademark Office on March 9, 2001.

The Commissioner is hereby authorized to charge any fees, which may be required, or credit any overpayment to Deposit Account No. 19-2386.

It is hereby respectfully submitted that the enclosed documents complete the filing of the above patent application and justify the filing date of January 31, 2001. Please telephone the undersigned at (512) 794-3600, if there are any questions.

LAW OFFICES OF
SKJERVEN MORRILL
MacPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

EXPRESS MAIL LABEL NO:

EL803198539US

Respectfully submitted,

Mary Jo Bertani

Mary Jo Bertani
Attorney for Applicants
Reg. No. 42,321

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US

CONFIRMATION NO. 5458

FORMALITIES LETTER

OC000000005843769

Mary Jo Bertani
SKJERVEN, MORRILL, MacPHERSON,
FRANKLIN & FRIEL LLP
25 Metro Drive, Suite 700
San Jose, CA 95110-1349

Date Mailed: 03/09/2001

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

This application has been accorded an Application Number and Filing Date. The application, however, is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given TWO MONTHS from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a)

The required item(s) identified below must be timely submitted to avoid abandonment:

- Substitute drawings in compliance with 37 CFR 1.84 because:
  - drawing sheets do not have the appropriate margin(s) (see 37 CFR 1.84(g)). Each sheet must include a top margin of at least 2.5 cm. (1 inch), a left side margin of at least 2.5 cm. (1 inch), a right side margin of at least 1.5 cm. (5/8 inch), and a bottom margin of at least 1.0 cm. (3/8 inch);

A copy of this notice MUST be returned with the reply.

Customer Service Center
Initial Patent Examination Division (703) 308-1202
PART 2 - COPY TO BE RETURNED WITH RESPONSE

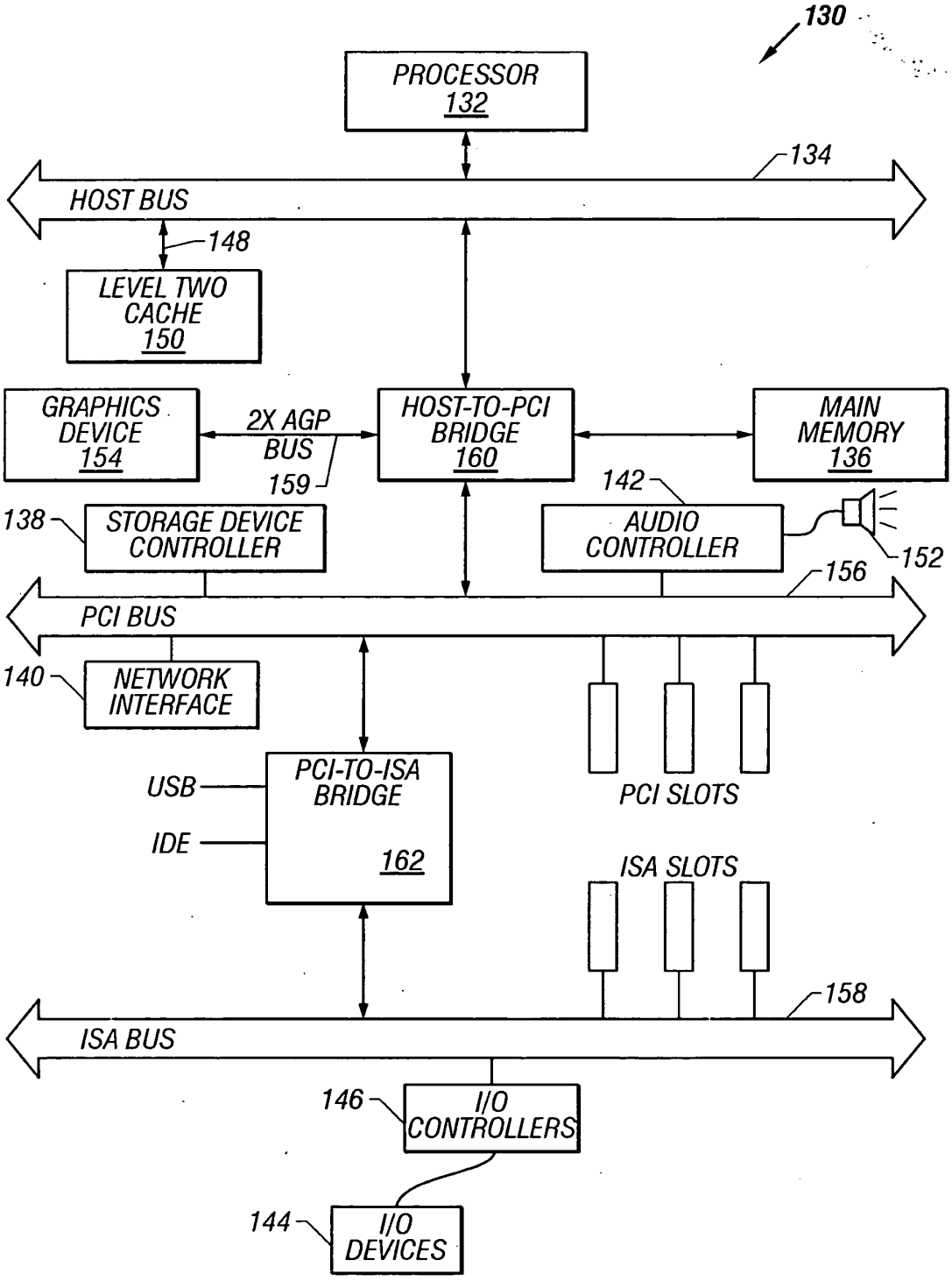

130
PROCESSOR
132
134
HOST BUS
148
LEVEL TWO CACHE
150
GRAPHICS DEVICE
154
2X AGP BUS
159
HOST-TO-PCI BRIDGE
160
MAIN MEMORY
136
142
138
STORAGE DEVICE CONTROLLER
AUDIO CONTROLLER
152
156
PCI BUS
140
NETWORK INTERFACE
USB
IDE
PCI-TO-ISA BRIDGE
162
PCI SLOTS
ISA SLOTS
158
ISA BUS
146
I/O CONTROLLERS
144
I/O DEVICES

FIG. 1

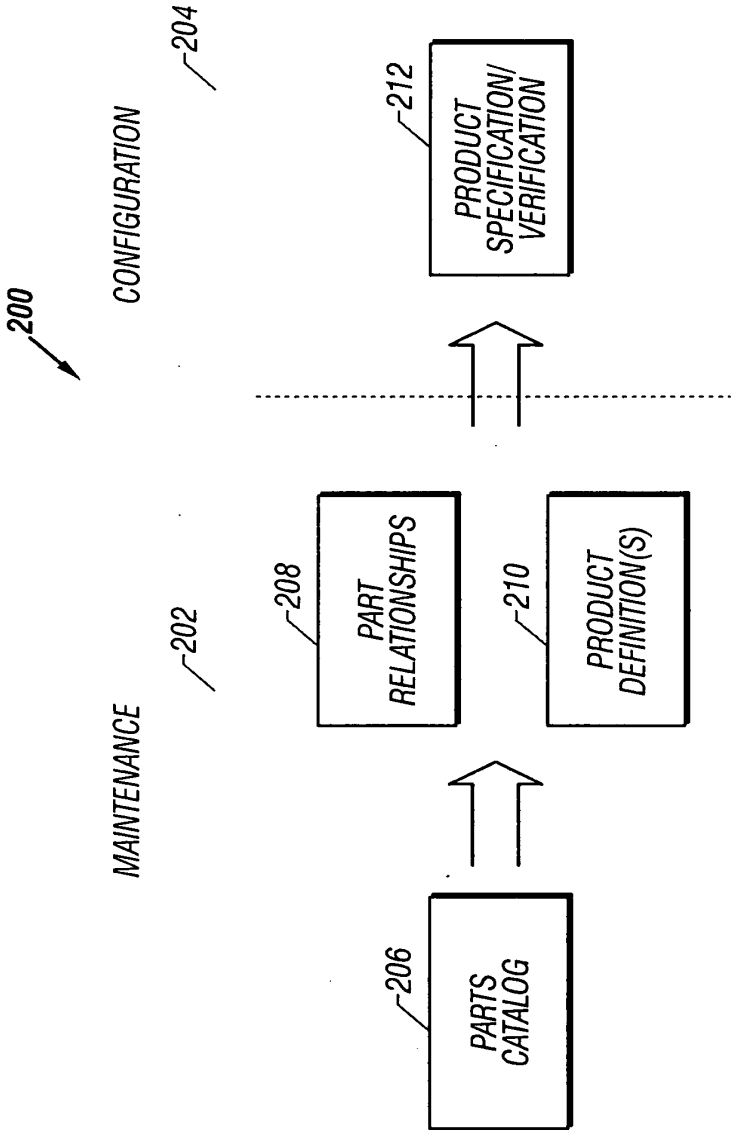


FIG. 2

3/6

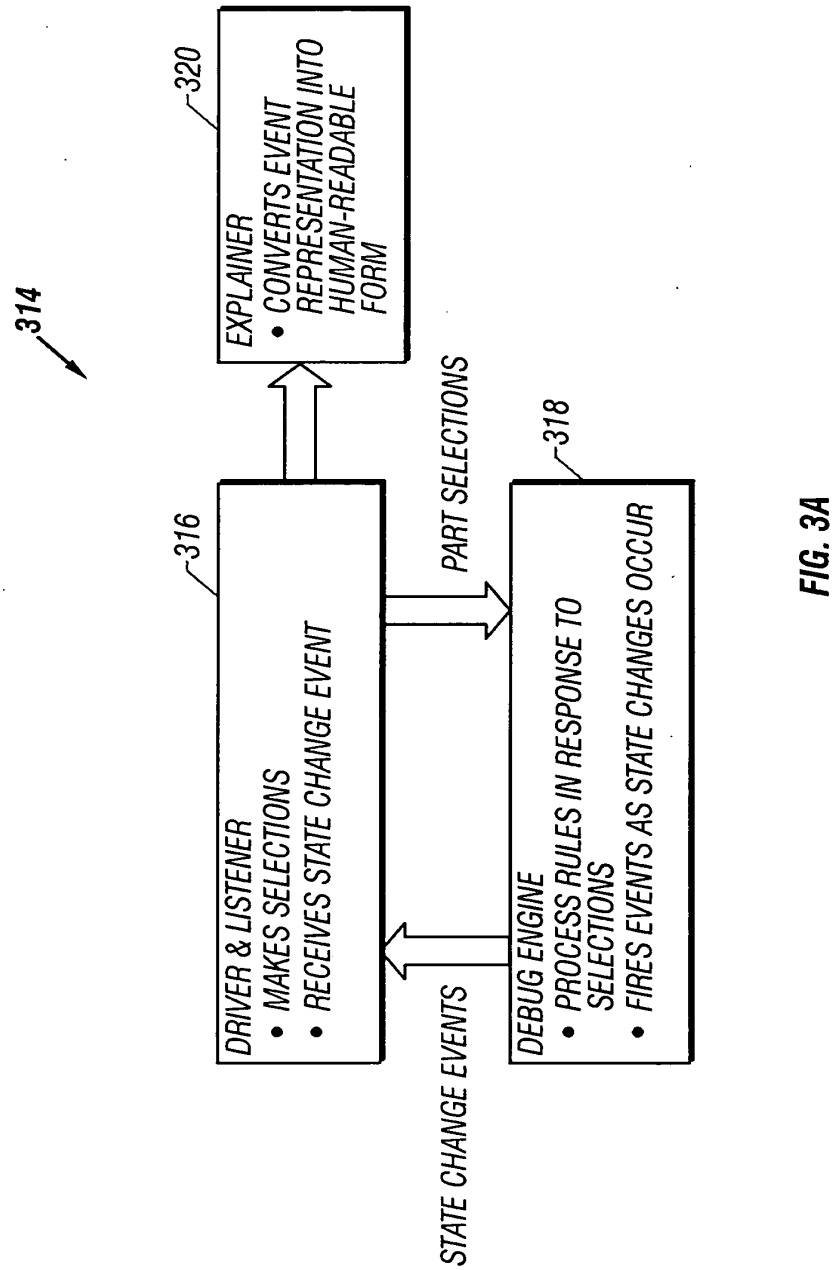


FIG. 3A

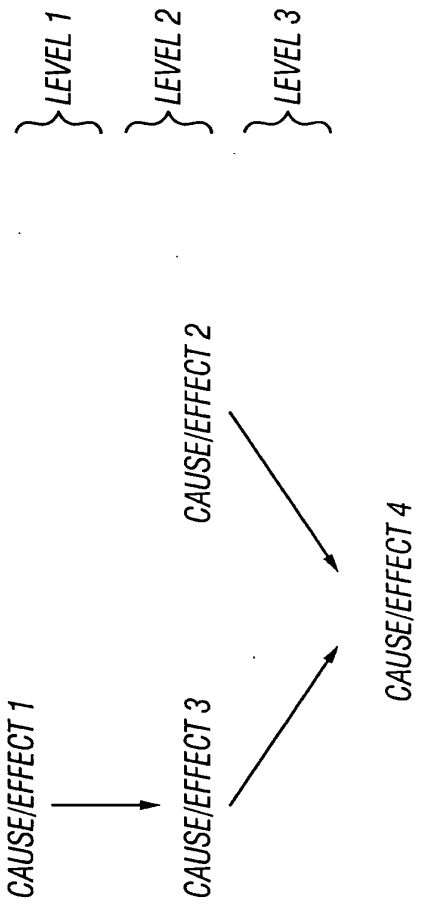


FIG. 3B

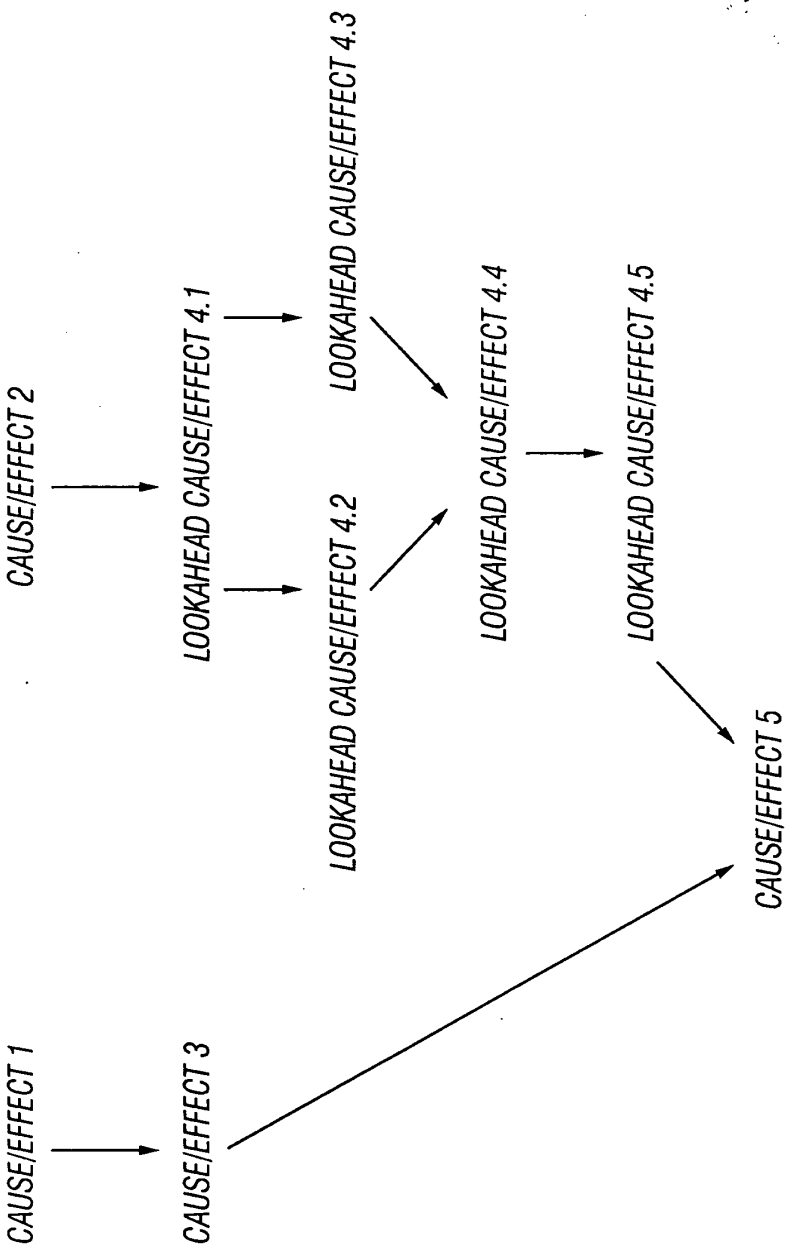

322
CAUSE/EFFECT 1
CAUSE/EFFECT 2
CAUSE/EFFECT 3
LOOKAHEAD CAUSE/EFFECT 4.1
LOOKAHEAD CAUSE/EFFECT 4.2
LOOKAHEAD CAUSE/EFFECT 4.3
LOOKAHEAD CAUSE/EFFECT 4.4
LOOKAHEAD CAUSE/EFFECT 4.5
CAUSE/EFFECT 5

FIG. 3C

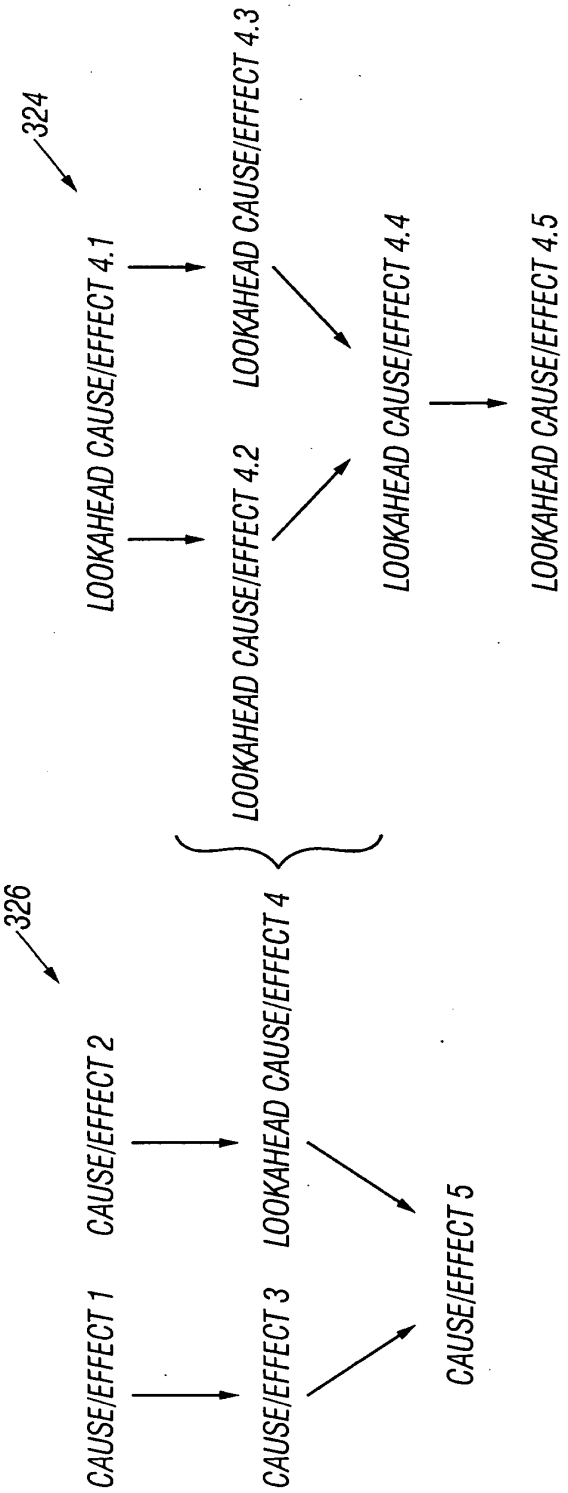


FIG. 3D

2121

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Kevin E. Gilpin et al.		
Assignee:	Trilogy Development Group		
Title:	RULES BASED CONFIGURATION ENGINE FOR A DATABASE		
Serial No.:	09/773,101	Filing Date:	January 31, 2001
Examiner:	Unknown	Group Art Unit:	2121

COMMISSIONER FOR PATENTS
Washington, D.C. 20231

POWER OF ATTORNEY BY ASSIGNEE OF ENTIRE INTEREST

Sir:

The above-identified assignee hereby revokes all powers of attorney previously given and appoints hereby the following attorney(s) and/or agent(s) to prosecute the above-identified application and to transact all business in the United States Patent and Trademark Office in connection therewith: Stephen A. Terrile (32,946), Gary W. Hamilton (31,834). Robert W. Holland (40,020), Michael R. Cannati (34,791); and Kent B. Chambers (38,893).

Please address all correspondence and telephone calls regarding this application to:

Kent B. Chambers
HAMILTON & TERRILE, LLP
P.O. Box 203518
Austin, Texas 78720
Telephone: (512) 338-9100
Facsimile: (512) 345-7225

ASSIGNEE CERTIFICATION UNDER 37 CFR 3.73(b)

The undersigned representative of the above-identified assignee certifies that the above-identified assignee is the assignee of the entire right, title and interest in the above-identified

patent application by virtue of a chain of title from the inventor(s) of the above-identified patent application to the above-identified assignee as shown below:

1. From: Kevin E. Gilpin, Adam R. Stein To: Trilogy Development Group. The document was recorded in the Patent and Trademark Office at Reel 011537, Frame 0965, or for which a copy thereof is attached.

The undersigned (whose title is supplied below) is empowered to sign this certificate on behalf of the above-identified assignee.

Trilogy Development Group, Inc. (Assignee)

By: _____

Name: Lance A. Jones

Title: Vice-President & General Counsel

Date: 5/29/02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Kevin E. Gilpin et al.		
Assignee:	Trilogy Development Group		
Title:	RULES BASED CONFIGURATION ENGINE FOR A DATABASE		
Serial No.:	09/773,101	Filing Date:	January 31, 2001
Examiner:	Unknown	Group Art Unit:	2121

NOTICE OF CHANGE OF ADDRESS

COMMISSIONER FOR PATENTS
Washington, D. C. 20231

Sir:

Please direct all correspondence in the above-identified patent to the undersigned at this address:

c/o HAMILTON & TERRILE, LLP
P.O. Box 203518
Austin, Texas 78720
Telephone: (512) 338-9100
Facsimile: (512) 345-7225

Respectfully submitted,

Kent B. Chambers
Attorney for Applicant(s)
Reg. No. 38,839

T0001

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US

CONFIRMATION NO. 5458

OC000000008531166

24251
SKJERVEN MORRILL LLP
25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110

Date Mailed: 07/29/2002

NOTICE REGARDING POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/17/2002.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

SANDRA R RANGE
2800 (703) 308-5850

OFFICE COPY

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US

CONFIRMATION NO. 5458

Kent B. Chambers
Hamilton and Terrile, LLP
P O Box 203518
Austin, TX 78720

OC000000008531173

Date Mailed: 07/29/2002

NOTICE REGARDING POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/17/2002.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

SANDRA R RANGE
2800 (703) 308-5850

OFFICE COPY

95 **Product assembly or manufacturing:**
This subclass is indented under 90. Subject matter wherein the data processing system or calculating computer controls, monitors, or manages the sequential operations of a production process.

SEE OR SEARCH CLASS:

12, Boot and Shoe Making, subclass 142 for the process of boot and shoe making.

29, Metal Working, appropriate subclass, especially subclasses 592 through 559for method of mechanical manufacture.

53, Package Making, appropriate subclass for an apparatus for and method of encompassing, encasing, or completely surrounding goods or materials with a cover made from sheet material stock.

56, Harvesters, appropriate subclass for a means of severing crops which grow above the surface of the ground, without disturbing the soil, and a means for gathering the same from the field after they are severed.

59, Chain, Staple, and Horseshoe Making, appropriate subclass for the manufacture as it relates to chains, staples, or horseshoes.

79, Button Making, appropriate subclass for machines or processes for making buttons.

118, Coating Apparatus, subclasses 663 through 714for the control of an apparatus for applying or obtaining a surface coating on a base or for impregnating base materials.

142, Wood Turning, appropriate subclasses for wood turning.

144, Woodworking, subclasses 329 through 381for woodworking processes.

156, Adhesive Bonding and Miscellaneous Chemical Manufacture, subclasses 1 through 344for a manufacturing process or apparatus including a step of adhesively bonding parts together or the manufacture of articles of commerce in which one of the manufacturing steps includes a chemical reaction.

162, Paper Making and Fiber Liberation, subclasses 100 through 231for the process of paper making.

163, Needle and Pin Making, subclasses 1 through 5for needle or pin manufacturing.

216, Etching a Substrate: Processes, appropriate subclasses for the manufacturing of a substrate by etching.

234, Selective Cutting (e.g., Punching), appropriate subclass for a method or apparatus for a selective cutting process.

264, Plastic and Nonmetallic Article Shaping or Treating: Processes, appropriate subclass.

300, Brush, Broom, and Mop Making, subclass 21 for the process of manufacturing.

340, Communications: Electrical, subclasses 3.1 through 3.9for selective electrical communication having monitoring in addition to control (e.g., supervisory).

412, Bookbinding: Process and Apparatus, subclasses 1 through 8for the process of bookbinding.

427, Coating Processes, appropriate subclass for applying or obtaining a coating or surface.

96 **Integrated system (Computer Integrated Manufacturing (CIM)):**
This subclass is indented under 95. Subject matter wherein the sequential operations of multiple manufacturing processes are interconnected by a host management system (e.g., Production Integrated Processing Equipment (PIPE), cluster tools, etc.).

97 **Design or planning:**
This subclass is indented under 95. Subject matter wherein the calculating computer or data processing system analyzes, prioritizes, or modifies input data to arrange the sequential operations for product manufacturing or to configure a product.

SEE OR SEARCH CLASS:

703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclass 1 for a data processing system or calculating computer designed for or utilized in structural design as it relates to mechanical engineering.

706, Data Processing: Artificial Intelligence, cross-reference art collections 919+ related to the designing of objects, plan preparation, program preparation, computer aided design (i.e., CAD), or computer aided software engineering (i.e., CASE).

716, Data Processing: Design and Analysis of Circuit or Semiconductor Mask, subclasses 1 through 18for a data processing system or calculating computer designed for or utilized in the design and analysis of electrical components and circuits made up thereof.

98 **3-D product design (e.g., solid modeling):**
This subclass is indented under 97. Subject matter wherein the planned or designed structure is represented as a three dimensional image in two-dimensional space.

SEE OR SEARCH THIS CLASS, SUBCLASS:

163, for 3-D sculpturing in the machining process.

SEE OR SEARCH CLASS:

345, Computer Graphics Processing, Operator Interface Processing, and Selective Visual Display Systems, subclasses 418 through 475and subclass 607 for three-dimensional or perspective data processing for display presentation.

359, Optics: Systems (Including Communication) and Elements, subclass 458 for stereoscopic or three-dimensional imaging.

382, Image Analysis, subclass 154 for 3-D or stereo image analysis.

99 **Resource allocation:**
This subclass is indented under 97. Subject matter wherein the data processing system or calculating computer control the coordination and logistics of physical objects in a manufacturing process.

SEE OR SEARCH THIS CLASS, SUBCLASS:

28 through 55, wherein a control seeks to optimize a system"s performance criterion (e.g., efficiency, consumption, or profit).

SEE OR SEARCH CLASS:

705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, subclass 8 for allocation of resources or scheduling an administrative function by an automated business or management system.

100 **Job scheduling:**
This subclass is indented under 99. Subject matter wherein the coordination of the physical object is controlled by a system constraint (e.g., time, machine availability, etc.).

SEE OR SEARCH CLASS:

234, Selective Cutting (e.g., Punching), subclasses 23 through 24for a means to detect the order of occurrence of input data.

101 **Priority ordering:**
This subclass is indented under 100. Subject matter wherein the data processing system or calculating computer operates in a supervisory mode to order the sequential operations.

102 **Job release determination:**
This subclass is indented under 100. Subject matter wherein the data processing system or calculating computer monitors and controls the sequence of manufacturing operations based on task output.

103 **Constraints or rules:**
This subclass is indented under 97. Subject matter wherein the data processing system or calculating computer monitors and controls the sequence of manufacturing operations based on a set of operating rules or regulations.

104 **Knowledge based (e.g., expert system):**
This subclass is indented under 103. Subject matter wherein the data processing system or calculating computer generates, monitors, modifies, or controls the sequential manufacturing operations using historical data to infer a result.

SEE OR SEARCH THIS CLASS, SUBCLASS:

49, for an expert system (e.g., knowledge based) control system.

SEE OR SEARCH CLASS:

706, Data Processing: Artificial Intelligence, subclass 14 for adaptive systems, cross-reference art collections 902-934 for art related to the applications of artificial intelligence, in particular cross reference art collection 904 wherein artificial intelligence is used in applications related to manufacturing and machines.

105 **Rework or engineering change:**
This subclass is indented under 97. Subject matter wherein the data processing system or calculating computer monitors and controls the sequence of manufacturing operations based on engineering or manufacturing changes.

106 **Material requirement:**
This subclass is indented under 97. Subject matter wherein the data processing system or calculating computer monitors and controls the sequence of manufacturing operations based on the necessary construction components.

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office

ASSISTANT SECRETARY AND COMMISSIONER
OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

CHANGE OF ADDRESS/POWER OF ATTORNEY

FILE LOCATION 2FC1 SERIAL NUMBER 09773101 PATENT NUMBER

THE CORRESPONDENCE ADDRESS HAS BEEN CHANGED TO CUSTOMER # 24251

THE PRACTITIONERS OF RECORD HAVE BEEN CHANGED TO CUSTOMER # 24251

ON 07/05/02 THE ADDRESS OF RECORD FOR CUSTOMER NUMBER 24251 IS:

SKJERVEN MORRILL MACPHERSON LLP
25 METRO DRIVE
SUITE 700
SAN JOSE CA 95110

AND THE PRACTITIONERS OF RECORD FOR CUSTOMER NUMBER 24251 ARE:

24423	24486	25875	29089	29545	32892	33003	33938	34250	34691
34763	36320	37119	38321	39880	40218	40350	40494	40684	41008
41711	42117	42268	42381	42406	42622	43339	43817	44040	44047
44702	44881	44940	45591	45754	45906	45983	46030	46091	46341
46930	47026	47063	47548	48049	48375	48562			

PTO INSTRUCTIONS: PLEASE TAKE THE FOLLOWING ACTION WHEN THE CORRESPONDENCE ADDRESS HAS BEEN CHANGED TO CUSTOMER NUMBER: RECORD, ON THE NEXT AVAILABLE CONTENTS LINE OF THE FILE JACKET, 'ADDRESS CHANGE TO CUSTOMER NUMBER'. LINE THROUGH THE OLD ADDRESS ON THE FILE JACKET LABEL AND ENTER ONLY THE 'CUSTOMER NUMBER' AS THE NEW ADDRESS. FILE THIS LETTER IN THE FILE JACKET. WHEN ABOVE CHANGES ARE ONLY TO FEE ADDRESS AND/OR PRACTITIONERS OF RECORD, FILE LETTER IN THE FILE JACKET.
THIS FILE IS ASSIGNED TO GAU 2857.

L Number	Hits	Search Text	DB	Time stamp
1	3	(("5,809,212") or ("5,603,031") or ("5,452,239")).PN.	USPAT; US-PGPUB	2003/12/13 19:20

Your search matched **[0]** of **[989552]** documents.

You may refine your search by editing the current search expression or entering a new one the text box. Then click search Again.

((test)and (configuration engine)) and(rule)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Your search matched **[0]** of **[987604]** documents.

You may refine your search by editing the current search expression or entering a new one the text box. Then click search Again.

((test)and (configuration engine))

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Your search matched **83** of **987604** documents.

A maximum of **83** results are displayed, **25** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text
Then click **Search Again**.

((test)and (configuration)) and rule

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **Process induced damages from various integrated circuit interconnect designs - limitations of antenna rule under practical integrated circuit la practice**
Mercier, J.; Dao, T.; Flechner, H.; Jean, B.; Oscar, D.B.; Aum, P.K.;
Plasma- and Process-Induced Damage, 2003 8th International Symposium , 24-2003
Page(s): 162 -167

[Abstract] [PDF Full-Text (584 KB)] **IEEE CNF**

2 **Analyzing the mechanical strength of SMT attached solder joints**
Chen, W.;
Electronic Manufacturing Technology Symposium, 1989, Proceedings. Seventh IEEE/CHMT International , 25-27 Sept. 1989
Page(s): 61 -69

[Abstract] [PDF Full-Text (468 KB)] **IEEE CNF**

3 **Automated schematic capture with the USC51 embedded microcontrol**
Aleman, E.M.; Couleur, J.L.;
Euro ASIC '90 , 29 May-1 June 1990
Page(s): 461 -465

[Abstract] [PDF Full-Text (368 KB)] **IEEE CNF**

4 **A fragmented register architecture and test advisor for BIST**
Illman, R.J.; Traynor, D.J.;

European Design and Test Conference, 1994. EDAC, The European Conference o Design Automation. ETC European Test Conference. EUROASIC, The European E ASIC Design, Proceedings. , 28 Feb.-3 March 1994
Page(s): 124 -129

[Abstract] [PDF Full-Text (304 KB)] **IEEE CNF**

5 **Issues in providing expert advice for users of a particle-in-cell simulat code**
Gladd, N.T.; Verboncoeur, J.P.;
Plasma Science, 1995. IEEE Conference Record - Abstracts., 1995 IEEE Internat Conference on , 5-8 June 1995
Page(s): 244 -245

[Abstract] [PDF Full-Text (180 KB)] **IEEE CNF**

6 **A neuro-chip with temporal learning: test results for signal/shape generation**
Salam, F.M.;
Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilo Conference on , Volume: 2 , 2-5 Nov. 1997
Page(s): 1141 -1145 vol.2

[Abstract] [PDF Full-Text (420 KB)] **IEEE CNF**

7 **The analysis and design of distributed detection based on star configu**

Yin Chengyou; Xu Shanjia; Wang Dongjin;
Aerospace and Electronics Conference, 1998. NAECON 1998. Proceedings of the 1998 National , 13-17 July 1998
Page(s): 492 -497

[Abstract] [PDF Full-Text (448 KB)] **IEEE CNF**

8 **Application of adaptive control to gold processing plant**
Burdett, B.W.; McKay, J.; Toronto, T.W.; Lampshire, D.; Ynchausti, R.A.; VanRip
Advanced Process Control Applications for Industry Workshop, 1999. IEEE Indus Applications Society , 29-30 April 1999
Page(s): 65 -74

[Abstract] [PDF Full-Text (540 KB)] **IEEE CNF**

9 **Distributed Bayesian hypothesis testing with distributed data fusion**
Chair, Z.; Varshney, P.K.;

Systems, Man and Cybernetics, IEEE Transactions on , Volume: 18 Issue: 5 , Se 1988
Page(s): 695 -699

[Abstract] [PDF Full-Text (312 KB)] **IEEE JNL**

10 **Performance evaluation of a cellular base station multibeam antenna**
Yingjie Li; Feuerstein, N.J.; Reudink, D.O.;
Vehicular Technology, IEEE Transactions on , Volume: 46 Issue: 1 , Feb. 1997
Page(s): 1 -9

[Abstract] [PDF Full-Text (192 KB)] **IEEE JNL**

11 **Airport surface collision warning system implementation**
Ianniello, J.W.; Kruczek, R.M.;
Vehicle Navigation and Information Systems Conference, 1993., Proceedings of IEEE-IEE , 12-15 Oct. 1993
Page(s): 742 -746

[Abstract] [PDF Full-Text (624 KB)] **IEEE CNF**

12 **Airport surface collision warning system implementation**
Ianniello, J.W.; Kruczek, R.M.;
Digital Avionics Systems Conference, 1993. 12th DASC., AIAA/IEEE , 25-28 Oct.
Page(s): 425 -429

[Abstract] [PDF Full-Text (492 KB)] **IEEE CNF**

13 **Architecture of the Texas A&M Autonomous Underwater Vehicle Cont**
Barnett, D.; McClaran, S.; Nelson, E.; McDermott, M.; Williams, G.;
Autonomous Underwater Vehicle Technology, 1996. AUV '96., Proceedings of the Symposium on , 2-6 June 1996
Page(s): 231 -237

[Abstract] [PDF Full-Text (672 KB)] **IEEE CNF**

14 **Implementation of ANN-based sensorless induction motor drives**
Vas, P.; Stronach, A.F.; Rashed, M.; Neuroth, M.;
Electrical Machines and Drives, 1999. Ninth International Conference on (Conf. P No. 468) , 1-3 Sept. 1999
Page(s): 329 -333

[Abstract] [PDF Full-Text (264 KB)] **IEE CNF**

15 **Initial development of a genetic algorithm to automate system implementation in a novel electronic packaging technology**
Larcombe, S.P.; Prendergast, D.J.; Thacker, N.A.; Ivey, P.A.;
Genetic Algorithms In Engineering Systems:Innovations And Applications, 1997 GALESIA 97. Second International Conference On (Conf. Publ. No. 446) , 2-4 Se 1997
Page(s): 486 -491

[Abstract] [PDF Full-Text (592 KB)] **IEE CNF**

16 **The use of a knowledge-based system in power electronic circuit faul diagnosis**
Renfrew, A.C.; Tian, J.X.;
Power Electronics and Applications, 1993., Fifth European Conference on , 13-16 1993
Page(s): 57 -62 vol.7

[Abstract] [PDF Full-Text (284 KB)] **IEE CNF**

17 **Implementation of BIST in 100 k gate ASICS**
Illman, R.; Traynor, D.;
Testing-the Gordian Knot of VLSI Design, IEE Colloquium on , 28 May 1993
Page(s): 2/1 -2/4

[Abstract] [PDF Full-Text (184 KB)] **IEE CNF**

18 **Software requirements for railway signalling systems**
Short, R.C.;
Software Requirements for High Integrity Systems, IEE Colloquium on , 10 Nov
Page(s): 4/1 -4/3

[Abstract] [PDF Full-Text (104 KB)] **IEE CNF**

19 **Combining few neural networks for effective secondary structure pre**
Guimaraes, K.S.; Melo, J.C.B.; Cavalcanti, G.D.C.;
Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium o 10-12 March 2003
Page(s): 415 -420

[Abstract] [PDF Full-Text (308 KB)] **IEEE CNF**

20 **Evaluation of an isolated word recognizer in talker-dependent and**

talker-independent modes using a large telephone-band data base
Rosenberg, A.; Shipley, K.;
Acoustics, Speech, and Signal Processing, IEEE International Conference on ICA '84. , Volume: 9 , Mar 1984
Page(s): 348 -351

[Abstract] [PDF Full-Text (120 KB)] **IEEE CNF**

21 **A model for change management**
Kramer, J.; Magee, J.;
Distributed Computing Systems in the 1990s, 1988. Proceedings., Workshop on Future Trends of , 14-16 Sept. 1988
Page(s): 286 -295

[Abstract] [PDF Full-Text (828 KB)] **IEEE CNF**

22 **The use of threshold logic to improve performance in scheduling a manufacturing process**
Farley, E.T.; Haworth, D.A.;
IEEE Region 5 Conference, 1988: 'Spanning the Peaks of Electrotechnology' , 21 March 1988
Page(s): 11 -14

[Abstract] [PDF Full-Text (216 KB)] **IEEE CNF**

23 **Criteria and designs for surge couplers and back-filters**
Richman, P.;
Electromagnetic Compatibility, 1989. IEEE 1989 National Symposium on , 23-25 1989
Page(s): 202 -207

[Abstract] [PDF Full-Text (504 KB)] **IEEE CNF**

24 **An investigation of neural networks for F-16 fault diagnosis. I. System description**
McDuff, R.J.; Simpson, P.K.; Gunning, D.;
AUTOTESTCON '89. IEEE Automatic Testing Conference. The Systems Readiness Technology Conference. Automatic Testing in the Next Decade and the 21st Cen Conference Record. , 25-28 Sept. 1989
Page(s): 351 -357

[Abstract] [PDF Full-Text (496 KB)] **IEEE CNF**

25 **Interfacing to boundary scan chips for system level BIT**

Turino, J.;
AUTOTESTCON '89. IEEE Automatic Testing Conference. The Systems Readiness Technology Conference. Automatic Testing in the Next Decade and the 21st Cen Conference Record. , 25-28 Sept. 1989
Page(s): 310 -313

[Abstract] [PDF Full-Text (224 KB)] **IEEE CNF**

1 2 3 4 [Next]

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ | Terms | Back to Top

Your search matched **83** of **987604** documents.

A maximum of **83** results are displayed, **25** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text
Then click **Search Again**.

((test)and (configuration)) and rule

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

26 **SUPER-GRIP: an expert system for grasping boxes**
Pelletier, M.; Hayward, V.;
Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE Internation Conference on , 14-17 Nov. 1989
Page(s): 510 -515 vol.2

[Abstract] [PDF Full-Text (412 KB)] **IEEE CNF**

27 **Acoustic-phonetic module for continuous Slovene speech recognition**
Mihelic, F.; Gyergyek, L.; Pavesic, N.;
Electrotechnical Conference, 1989. Proceedings. 'Integrating Research, Industry Education in Energy and Communication Engineering', MELECON '89., Mediterra 11-13 April 1989
Page(s): 249 -252

[Abstract] [PDF Full-Text (312 KB)] **IEEE CNF**

28 **A 512 kb/5 ns BiCMOS RAM with 1 kG/150 ps logic gate array**
Odaka, M.; Nakamura, K.; Eno, K.; Ogiue, K.; Saito, O.; Ikeda, T.; Hirao, M.; H H.;
Solid-State Circuits Conference, 1989. Digest of Technical Papers. 36th ISSCC., IEEE International , 15-17 Feb. 1989
Page(s): 28 -29, 279

[Abstract] [PDF Full-Text (272 KB)] **IEEE CNF**

29 **Using executable specification languages for interface checking of lar**

distributed systems
Liu, L.M.; Prywes, N.S.;
System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii Interna
Conference on , Volume: ii , 2-5 Jan. 1990
Page(s): 174 -182 vol.2

[Abstract] [PDF Full-Text (556 KB)] **IEEE CNF**

30 **Test structures and finite element models for chip stress and plastic package reliability**
Pendse, R.; Demmin, J.;
Microelectronic Test Structures, 1990. ICMTS 1990. Proceedings of the 1990 International Conference on , 5-7 March 1990
Page(s): 155 -160

[Abstract] [PDF Full-Text (392 KB)] **IEEE CNF**

31 **Towards a platform for simulating the state-driven real-time control o flexible manufacturing cell**
Del Sordo, M.; Pascale, A.; Piscitelli, G.;
Computer Integrated Manufacturing, 1990., Proceedings of Rensselaer's Second International Conference on , 21-23 May 1990
Page(s): 273 -280

[Abstract] [PDF Full-Text (396 KB)] **IEEE CNF**

32 **Prediction of protein folding using the shift-learning method with a la scale neural network**
Poliac, M.O.; Wilcox, G.L.; Xin, Y.; Carmeli, T.; Liebman, M.;
Neural Networks, 1991. 1991 IEEE International Joint Conference on , 18-21 No

Page(s): 1323 -1399 vol.2

[Abstract] [PDF Full-Text (416 KB)] **IEEE CNF**

33 **A flexible formant synthesizer**
Lalwani, A.L.; Childers, D.G.;
Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 Internationa Conference on , 14-17 April 1991
Page(s): 777 -780 vol.2

[Abstract] [PDF Full-Text (488 KB)] **IEEE CNF**

34 **Simulation-based planning of robot tasks in flexible manufacturing**

Rozenblit, J.W.; Jacak, W.;
AI, Simulation, and Planning in High Autonomy Systems, 1991. 'Integrating Qua and Quantitative System Knowledge'., Proceedings of the Second Annual Confer , 1-2 April 1991
Page(s): 166 -173

[Abstract] [PDF Full-Text (568 KB)] **IEEE CNF**

35 **Automating testability analysis of analog circuits and systems**
Kuhns, F.;
AUTOTESTCON '92. IEEE Systems Readiness Technology Conference, Conferenc Record , 21-24 Sept. 1992
Page(s): 225 -231

[Abstract] [PDF Full-Text (636 KB)] **IEEE CNF**

36 **The effects of segmentation on back-propagation networks**
Calvert, D.; Stacey, D.;
Neural Networks, 1992. IJCNN., International Joint Conference on , Volume: 1 , June 1992
Page(s): 907 -913 vol.1

[Abstract] [PDF Full-Text (456 KB)] **IEEE CNF**

37 **Learning fuzzy rules from artificial neural nets**
Textor, W.; Wessel, S.; Hoffgen, K.-U.;
CompEuro '92 . 'Computer Systems and Software Engineering',Proceedings. , 4- 1992
Page(s): 121 -126

[Abstract] [PDF Full-Text (472 KB)] **IEEE CNF**

38 **Ultrasonic echo simulator for mobile robots**
Tsuzuki, F.; Sasaki, K.;
Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on , Volume: 2 , 26-30 July 1993
Page(s): 979 -985 vol.2

[Abstract] [PDF Full-Text (560 KB)] **IEEE CNF**

39 **Design and VLSI implementation of a novel concurrent 16-bit multiplier-accumulator for DSP applications**
Poornaiah, D.V.; Haribabu, R.; Ahmad, M.O.;
Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE Interna

Conference on , Volume: 1 , 27-30 April 1993
Page(s): 385 -388 vol.1

[Abstract] [PDF Full-Text (296 KB)] **IEEE CNF**

---

40 **Protein alpha -helix region prediction based on stochastic-rule learnin**
Mamitsuka, H.; Yamanishi, K.;
System Sciences, 1993, Proceeding of the Twenty-Sixth Hawaii International Conference on , Volume: i , 5-8 Jan 1993
Page(s): 659 -668 vol.1

[Abstract] [PDF Full-Text (858 KB)] **IEEE CNF**

---

41 **A computer hardware configuration expert system: Examination of its software reliability**
Mori, N.; Mori, S.; Matsumoto, A.;
Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on , March 1993
Page(s): 306 -311

[Abstract] [PDF Full-Text (400 KB)] **IEEE CNF**

---

42 **A Statecharts implementation to run simulation of control dealing wit parameterized states**
Marty, J.C.; Sartor, M.;
Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Hu Conference Proceedings., International Conference on , 17-20 Oct. 1993
Page(s): 240 -245 vol.2

[Abstract] [PDF Full-Text (516 KB)] **IEEE CNF**

---

43 **Fuzzy logic for depth control of unmanned undersea vehicles**
DeBitetto, P.A.;
Autonomous Underwater Vehicle Technology, 1994. AUV '94., Proceedings of the Symposium on , 19-20 July 1994
Page(s): 233 -241

[Abstract] [PDF Full-Text (744 KB)] **IEEE CNF**

---

44 **Force directed self-organizing maps for L-shaped cell placement usin learning rule**
Ray-I Chang; Pei-Yung Hsiao;
Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 19 IEEE International Conference on , Volume: 5 , 27 June-2 July 1994

Page(s): 3381 -3386 vol.5

[Abstract] [PDF Full-Text (408 KB)] **IEEE CNF**

45 **Life loss of insulated power cables: an integrative criterium to improv ANSI/IEEE and the CENELEC/IEC method for overload protection**
Parise, G.; Rubino, G.; Ricci, M.;
Industry Applications Conference, 1996. Thirty-First IAS Annual Meeting, IAS '9 Conference Record of the 1996 IEEE , Volume: 4 , 6-10 Oct. 1996
Page(s): 2449 -2454 vol.4

[Abstract] [PDF Full-Text (544 KB)] **IEEE CNF**

46 **Using genetic algorithms to automate system implementation in a no three-dimensional packaging technology**
Larcombe, S.P.; Prendergast, D.J.; Thacker, N.A.; Ivey, P.A.;
Computer Design: VLSI in Computers and Processors, 1996. ICCD '96. Proceedi 1996 IEEE International Conference on , 7-9 Oct. 1996
Page(s): 274 -279

[Abstract] [PDF Full-Text (672 KB)] **IEEE CNF**

47 **A methodological approach for improvement of vacuum-insulated HV bushings**
Lupo, G.; Petrarca, C.; Egiziano, L.; Spagnuolo, G.; Tucci, V.;
Discharges and Electrical Insulation in Vacuum, 1996. Proceedings. ISDEIV., XV International Symposium on , Volume: 1 , 21-26 July 1996
Page(s): 552 -556 vol.1

[Abstract] [PDF Full-Text (404 KB)] **IEEE CNF**

48 **DC-current-free low-power A/D converter circuitry using dynamic lat comparators with divided-capacitance voltage reference**
Kotani, K.; Shibata, T.; Ohmi, T.;
Circuits and Systems, 1996. ISCAS '96., 'Connecting the World'., 1996 IEEE International Symposium on , Volume: 4 , 12-15 May 1996
Page(s): 205 -208 vol.4

[Abstract] [PDF Full-Text (504 KB)] **IEEE CNF**

49 **Fully self-aligned $6F^2$ cell technology for low cost 1Gb DRAM**
Aoki, M.; Ozaki, T.; Yamada, T.; Kawaguchiya, H.; Ishibashi, Y.; Hamamoto, T.;
VLSI Technology, 1996. Digest of Technical Papers. 1996 Symposium on , 11-13 1996

Page(s): 22 -23

[Abstract] [PDF Full-Text (224 KB)] **IEEE CNF**

---

50 **Tritium reduction and control in the vacuum vessel during TFTR outag decommissioning**

Blanchard, W.; Camp, R.; Carnevale, H.; Casey, M.; Collins, J.; Gentile, C.A.; G M.; Hosea, J.C.; Kalish, M.; Langford, J.; Langish, S.; Miller, D.; Nagy, A.; Pears G.G.; Raucci, R.; Rule, K.; Winston, J.;

Fusion Engineering, 1997. 17th IEEE/NPSS Symposium , Volume: 1 , 6-10 Oct. 1

Page(s): 297 -300 vol.1

[Abstract] [PDF Full-Text (400 KB)] **IEEE CNF**

[Prev] 1 2 3 4 [Next]

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Your search matched **83** of **987604** documents.

A maximum of **83** results are displayed, **25** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text
Then click **Search Again**.

((test)and (configuration)) and rule

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

51 **High-performance component software changes the rules for configu ATE**
Stern, P.;
AUTOTESTCON '97. 1997 IEEE Autotestcon Proceedings , 22-25 Sept. 1997
Page(s): 148 -152

[Abstract] [PDF Full-Text (444 KB)] **IEEE CNF**

52 **The development of PC-based highly model-mixed engine dispatching systems**
Chung-Hsien Kuo; Han-Pang Huang; Wei, K.C.; Tang, S.S.H.;
Emerging Technologies and Factory Automation Proceedings, 1997. ETFA '97., 1 International Conference on , 9-12 Sept. 1997
Page(s): 49 -54

[Abstract] [PDF Full-Text (732 KB)] **IEEE CNF**

53 **Parallel processing machine vision system for bare PCB inspection**
Ji-joong Hong; Kyung-ja Park; Kyung-gu Kim;
Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE , Volume: 3 , 31 Aug.-4 Sept. 1998
Page(s): 1346 -1350 vol.3

[Abstract] [PDF Full-Text (408 KB)] **IEEE CNF**

54 **On the identification of optimal cellular automata for built-in self-test sequential circuits**

Corno, F.; Gaudenzi, N.; Prinetto, P.; Reorda, M.S.;
VLSI Test Symposium, 1998. Proceedings. 16th IEEE , 26-30 April 1998
Page(s): 424 -429

[Abstract] [PDF Full-Text (72 KB)] **IEEE CNF**

---

55 **Determining redundancy requirements for memory arrays with critica analysis**
Segal, J.D.; Bakarian, S.; Colburn, J.E.; Kumar, M.; Hong, C.; Shubat, A.;
Memory Technology, Design and Testing, 1999. Records of the 1999 IEEE Intern Workshop on , 9-10 Aug. 1999
Page(s): 48 -53

[Abstract] [PDF Full-Text (120 KB)] **IEEE CNF**

---

56 **Building simulators for aerospace applications: processes, techniques choices and pitfalls**
Eccles, D.S.;
Aerospace Conference Proceedings, 2000 IEEE , Volume: 11 , 18-25 March 2000
Page(s): 517 -527 vol.1

[Abstract] [PDF Full-Text (1136 KB)] **IEEE CNF**

---

57 **Cooperation between two omnidirectional perception systems for mo robot localization**
Clerentin, A.; Delahoche, L.; Brassart, E.;
Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RS International Conference on , Volume: 2 , 31 Oct.-5 Nov. 2000
Page(s): 1499 -1504 vol.2

[Abstract] [PDF Full-Text (500 KB)] **IEEE CNF**

---

58 **Policy trading**
Hanssen, O.; Eliassen, F.;
Distributed Objects and Applications, 2000. Proceedings. DOA '00. International Symposium on , 21-23 Sept. 2000
Page(s): 219 -227

[Abstract] [PDF Full-Text (744 KB)] **IEEE CNF**

---

59 **URC fuzzy modeling and simulation of gene regulation**
Sokhansanj, B.A.; Fitch, J.P.;
Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Ann International Conference of the IEEE , Volume: 3 , 25-28 Oct. 2001

Page(s): 2918 -2921 vol.3

[Abstract] [PDF Full-Text (530 KB)] **IEEE CNF**

60 **Yemanja-a layered event correlation engine for multi-domain server f**
Appleby, K.; Goldszmidt, G.; Steinder, M.;
Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on , 14-18 May 2001
Page(s): 329 -344

[Abstract] [PDF Full-Text (472 KB)] **IEEE CNF**

61 **Technology development for react and wind common coil magnets**
Escallier, J.; Anerella, M.; Cozzolino, J.; Ganetis, G.; Ghosh, A.; Gupta, R.; Harr M.; Marone, A.; Muratore, J.; Parker, B.; Sampson, W.; Wanderer, P.;
Particle Accelerator Conference, 2001. PAC 2001. Proceedings of the 2001 , Volu 18-22 June 2001
Page(s): 214 -216 vol.1

[Abstract] [PDF Full-Text (319 KB)] **IEEE CNF**

62 **Rules for a cellular automaton to model quantum-dot cellular automa**
Cole, T.; Lusth, J.C.;
Nanotechnology, 2001. IEEE-NANO 2001. Proceedings of the 2001 1st IEEE Con on , 28-30 Oct. 2001
Page(s): 391 -396

[Abstract] [PDF Full-Text (425 KB)] **IEEE CNF**

63 **SI and design considerations for Gbps PCBs in communication system**
Zhen Mu; Willis, K.;
Electrical Performance of Electronic Packaging, 2001 , 29-31 Oct. 2001
Page(s): 287 -290

[Abstract] [PDF Full-Text (297 KB)] **IEEE CNF**

64 **An adaptive weighted majority vote rule for combining multiple class**
De Stefano, C.; Della Cioppa, A.; Marcelli, A.;
Pattern Recognition, 2002. Proceedings. 16th International Conference on , Volu 11-15 Aug. 2002
Page(s): 192 -195 vol.2

[Abstract] [PDF Full-Text (359 KB)] **IEEE CNF**

65 **Intelligent system for control of a stepping motor drive using a hybri neuro-fuzzy approach**
Melin, P.; Castillo, O.;
Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE Internatio Symposium on , Volume: 1 , 8-11 July 2002
Page(s): 305 -309 vol.1

[Abstract] [PDF Full-Text (582 KB)] **IEEE CNF**

66 **Experimental investigations of digital signal processing techniques in FMCW radar for naval application**
Grzywacz, A.;
Microwaves, Radar and Wireless Communications, 2002. MIKON-2002. 14th International Conference on , Volume: 3 , 20-22 May 2002
Page(s): 757 -763 vol.3

[Abstract] [PDF Full-Text (581 KB)] **IEEE CNF**

67 **An evolutionary algorithm for classifier and combination rule selectio multiple classifier systems**
Sirlantzis, K.; Fairhurst, M.C.; Guest, R.M.;
Pattern Recognition, 2002. Proceedings. 16th International Conference on , Volu 11-15 Aug. 2002
Page(s): 771 -774 vol.2

[Abstract] [PDF Full-Text (341 KB)] **IEEE CNF**

68 **Switched reluctance motors with segmental rotors**
Mecrow, B.C.; Finch, J.W.; El-Kharashi, E.A.; Jack, A.G.;
Electric Power Applications, IEE Proceedings- , Volume: 149 Issue: 4 , July 2002
Page(s): 245 -254

[Abstract] [PDF Full-Text (1425 KB)] **IEE JNL**

69 **Applying artificial intelligence techniques to human-computer interfa**
Sonnenwald, D.H.;
Communications Magazine, IEEE , Volume: 26 Issue: 3 , March 1988
Page(s): 14 -20

[Abstract] [PDF Full-Text (544 KB)] **IEEE JNL**

70 **Thermal characteristics of TAB for small systems**
McNelis, B.; Buller, L.;

Components, Hybrids, and Manufacturing Technology, IEEE Transactions on [se
IEEE Trans. on Components, Packaging, and Manufacturing Technology, Part A,
Volume: 13 Issue: 4 , Dec. 1990
Page(s): 989 -997

[Abstract] [PDF Full-Text (792 KB)] **IEEE JNL**

71 **The evolving philosophers problem: dynamic change management**
Kramer, J.; Magee, J.;
Software Engineering, IEEE Transactions on , Volume: 16 Issue: 11 , Nov. 1990
Page(s): 1293 -1306

[Abstract] [PDF Full-Text (1312 KB)] **IEEE JNL**

72 **Distribution planning using a knowledge-based expert system**
Hsu, Y.-Y.; Chen, J.-L.;
Power Delivery, IEEE Transactions on , Volume: 5 Issue: 3 , July 1990
Page(s): 1514 -1519

[Abstract] [PDF Full-Text (380 KB)] **IEEE JNL**

73 **Sailing off the edge of the earth. . .again**
Greenbaum, J.R.;
Circuits and Devices Magazine, IEEE , Volume: 7 Issue: 3 , May 1991
Page(s): 11 -17

[Abstract] [PDF Full-Text (1356 KB)] **IEEE JNL**

74 **Inferring feasible assemblies from spatial constraints**
Thomas, F.; Torras, C.;
Robotics and Automation, IEEE Transactions on , Volume: 8 Issue: 2 , April 1992
Page(s): 228 -239

[Abstract] [PDF Full-Text (1004 KB)] **IEEE JNL**

75 **Robotic collision avoidance in a flexible assembly cell using a dynami knowledge base**
Manivannan, S.;
Systems, Man and Cybernetics, IEEE Transactions on , Volume: 23 Issue: 3 , Ma 1993
Page(s): 766 -782

[Abstract] [PDF Full-Text (1412 KB)] **IEEE JNL**

[Prev] 1 2 3 4 [Next]

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ | Terms | Back to Top

Your search matched **83** of **987604** documents.

A maximum of **83** results are displayed, **25** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text
Then click **Search Again**.

((test)and (configuration)) and rule

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

76 **Modeling of oxide breakdown from gate charging during resist ashing**
Fang, S.; Murakawa, S.; McVittie, J.P.;
Electron Devices, IEEE Transactions on , Volume: 41 Issue: 10 , Oct. 1994
Page(s): 1848 -1855

[Abstract] [PDF Full-Text (736 KB)] **IEEE JNL**

77 **Fuzzy logic for depth control of Unmanned Undersea Vehicles**
DeBitetto, P.A.;
Oceanic Engineering, IEEE Journal of , Volume: 20 Issue: 3 , July 1995
Page(s): 242 -248

[Abstract] [PDF Full-Text (644 KB)] **IEEE JNL**

78 **Wire-antenna designs using genetic algorithms**
Altshuler, E.E.; Linden, D.S.;
Antennas and Propagation Magazine, IEEE , Volume: 39 Issue: 2 , April 1997
Page(s): 33 -43

[Abstract] [PDF Full-Text (1092 KB)] **IEEE JNL**

79 **Hybrid optical-electrical overlay test structure**
Cresswell, M.W.; Allen, R.A.; Linholm, L.W.; Guthrie, W.F.; Penzes, W.B.; Gurne A.W.;
Semiconductor Manufacturing, IEEE Transactions on , Volume: 10 Issue: 2 , May
Page(s): 250 -255

[Abstract] [PDF Full-Text (156 KB)] **IEEE JNL**

80 **A-SMGCS routing and guidance functions**
Piazza, E.;
Aerospace and Electronic Systems Magazine, IEEE , Volume: 15 Issue: 7 , July 2
Page(s): 15 -23

[Abstract] [PDF Full-Text (628 KB)] **IEEE JNL**

81 **A neuron adaptive detecting approach of harmonic current for APF an realization of analog circuit**
Qun Wang; Ning Wu; Zhaoan Wang;
Instrumentation and Measurement, IEEE Transactions on , Volume: 50 Issue: 1
2001
Page(s): 77 -84

[Abstract] [PDF Full-Text (168 KB)] **IEEE JNL**

82 **Switching overvoltages in motor circuits**
Berth, M.; Kung, M.; Limbeek, E.F.D.E.;
Industry Applications, IEEE Transactions on , Volume: 37 Issue: 6 , Nov.-Dec. 2
Page(s): 1582 -1589

[Abstract] [PDF Full-Text (157 KB)] **IEEE JNL**

83 **Soft fault detection and isolation in analog circuits: some results and comparison between a fuzzy approach and radial basis function network**
Catelani, M.; Fort, A.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 51 Issue: 2
2002
Page(s): 196 -202

[Abstract] [PDF Full-Text (245 KB)] **IEEE JNL**

[Prev] 1 2 3 4

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9
Best 200 shown

1 Rule condition testing and action execution in Ariel
Eric N. Hanson
June 1992 ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD internati Volume 21 Issue 2
Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references,

This paper describes testing of rule conditions and execution of rule actions in is tightly coupled with query and update processing. Ariel rules can have cond and transitions. For testing rule conditions, Ariel makes use of a discriminatio structure for testing single-relation selection conditions efficiently, and a mod A-TREAT, ...

2 A predicate matching algorithm for database rule systems
Eric N. Hanson, Moez Chaabouni, Chang-Ho Kim, Yu-Wang Wang
May 1990 ACM SIGMOD Record , Proceedings of the 1990 ACM SIGMOD internatio Volume 19 Issue 2
Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references,

Forward-chaining rule systems must test each newly asserted fact against a c that match the fact. Expert system rule engines use a simple combination of matching. We introduce an algorithm for finding the matching predicates that algorithm when the number of predicates is large. We focus on equality and i domains. This algorithm is well ...

3 An algebraic approach to static analysis of active database rules
Elena Baralis, Jennifer Widom
September 2000 ACM Transactions on Database Systems (TODS), Volume 2
Full text available: pdf(391.93 KB) Additional Information: full citation, abstract, reference

Rules in active database systems can be very difficult to program due to the u rule processing. We provide static analysis techniques for predicting whether and whether rule execution is confluent (guaranteed to have a unique final st techniques for analyzing rules in active database systems. We improve consid providing analysis criter ...

Keywords: active database systems, confluence, database rule processing, da

4 Efficient tests for top-down termination of logical rules
Jeffrey D. Ullman, Allen Van Gelder
April 1988 Journal of the ACM (JACM), Volume 35 Issue 2
Full text available: pdf(2.32 MB) Additional Information: full citation, abstract, references, citin

Considered is the question of whether top-down (Prolog-like) evaluation of a terminate. The NAIL! system is designed to process programs consisting of lo of the program, the best from among many possible strategies for its evaluat essential that termination tests be fast. Thus, the “uniqueness” property is ...

5 Set-oriented constructs: from Rete rule bases to database systems
Douglas N. Gordin, Alexander J. Pasik
April 1991 ACM SIGMOD Record , Proceedings of the 1991 ACM SIGMOD internati Volume 20 Issue 2
Full text available: pdf(733.87 KB) Additional Information: full citation, references, c

6 Discrimination network for rule condition matching in object-oriented datab
Moez Chaabouni, Soon M. Chung
February 1995 Proceedings of the 1995 ACM symposium on Applied computin
Full text available: pdf(641.89 KB) Additional Information: full citation, references, ind

Keywords: discrimination network, object-oriented database rule system, patt

7 Active rules in deductive databases
John V. Harrison
December 1993 Proceedings of the second international conference on Information
Full text available: pdf(1.13 MB) Additional Information: full citation, references, index

8 A practical approach to static analysis and execution of rules in active data
Seung-Kyum Kim, Sharma Chakravarthy
January 1997 Proceedings of the sixth international conference on Information and
Full text available: pdf(1.20 MB) Additional Information: full citation, references, index te

9 Column: Generating consistent test data: restricting the search space by a
Andrea Neufeld, Guido Moerkotte, Peter C. Lockemann
April 1993 The VLDB Journal — The International Journal on Very Large D
Full text available: pdf(2.31 MB) Additional Information: full citation, abstract, refer

To address the problem of generating test data for a set of general consistenc approach: First the interdependencies between consistency constraints are ex on their basis. During its creation, the user may exert control. In essence, the restrict the search for consistent test databases. In the second step, the test approaches are pr ...

Keywords: consistency, design, logic, test data, validation

10 Poster papers: Construct robust rule sets for classification
Jiuyong Li, Rodney Topor, Hong Shen
July 2002 Proceedings of the eighth ACM SIGKDD international conference on Kn
Full text available: pdf(612.74 KB) Additional Information: full citation, abstract, refer

We study the problem of computing classification rule sets from relational dat made on test data with missing attribute values. Traditional classifiers perform as the training data because they tailor a training database too much. We intr more robust than another, that is, able to make more accurate predictions on show that the opti ...

Keywords: association rule, classification rule, data mining

11 Efficient mining of association rules in text databases
John D. Holt, Soon M. Chung
November 1999 Proceedings of the eighth international conference on Informatio
Full text available: pdf(1.09 MB) Additional Information: full citation, abstract, referen

In this paper, we propose two new algorithms for mining association rules bet characteristics of text databases are quite different from those of retail transa algorithms cannot handle text databases efficiently because of the large num counted. Two well-known mining algorithms, Apriori algorithm and Direct Has evaluated in the context of min ...

12 Automating software analysis and testing using a program transformation s
G. Kotik, L. Markosian
November 1989 ACM SIGSOFT Software Engineering Notes , Proceedings of the AC Software testing, analysis, and verification, Volume 14 Issue 8
Full text available: pdf(888.08 KB) Additional Information: full citation, abstract, reference

We describe an approach to software analysis and test generation that combin databases and parsers for capturing and representing software; pattern langu querying and analyzing a database of software; and transformation rules for a on the analysis results, and for automatically creating program “mutan coverage of the test cases. We pre ...

13 Beyond market baskets: generalizing association rules to correlations
Sergey Brin, Rajeev Motwani, Craig Silverstein
June 1997 ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD internati Volume 26 Issue 2
Full text available: pdf(1.59 MB) Additional Information: full citation, abstract, references,

One of the most well-studied problems in data mining is mining for associatio rules, whose significance is measured via support and confidence, are intende customer purchasing item A often also purchases item B.” Motivated b baskets and the association rules used with them, we develop the notion of m (generalizing associations ...

14 Space optimization in deductive databases
Divesh Srivastava, S. Sudarshan, Raghu Ramakrishnan, Jeffrey F. Naughton
December 1995 ACM Transactions on Database Systems (TODS), Volume 20
Full text available: pdf(3.22 MB) Additional Information: full citation, abstract, references, in

In the bottom-up evaluation of logic programs and recursively defined views o usually assumed to be stored until the end of the evaluation. Discarding facts considerably improve the efficiency of the evaluation: the space needed to ev costs of maintaining and accessing indices, and the cost of eliminating duplica evaluation method that is sound, compl ...

Keywords: bottom-up query evaluation deductive database systems, discardin

15 On the Complexity of Testing Implications of Functional and Join Dependei
David Maier, Yehoshua Sagiv, Mihalis Yannakakis
October 1981 Journal of the ACM (JACM), Volume 28 Issue 4
Full text available: pdf(1.08 MB) Additional Information: full citation, references, citings, index terms

16 Index support for rule activation
David A. Brant, Daniel P. Miranker
June 1993 ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD internati Volume 22 Issue 2
Full text available: pdf(888.85 KB) Additional Information: full citation, abstract, reference

Integrated rule and database systems are quickly moving from the research l However, the current generation of prototypes are designed to work with sma The problem of supporting large complex rule programs within database mana challenges. The basis for many of these challenges is providing support for ru the process of determining ...

17 Special issue on prototypes of deductive database systems: The aditi dedu
Jayen Vaghani, Kotagiri Ramamohanarao, David B. Kemp, Zoltan Somogyi, Pete
April 1994 The VLDB Journal — The International Journal on Very Large D
Full text available: pdf(2.67 MB) Additional Information: full citation, abstract, refer

Deductive databases generalize relational databases by providing support for is a deductive system based on the client-server model; it is inherently multi- on shared-memory multiprocessors. The back-end uses relational technology disk-based data and uses optimization algorithms especially developed for the involving recursion. The front ...

Keywords: implementation, logic, multi-user, parallelism, relational database

18 A statistical theory for quantitative association rules
Yonatan Aumann, Yehuda Lindell
August 1999 Proceedings of the fifth ACM SIGKDD international conference on Kno
Full text available: pdf(1.22 MB) Additional Information: full citation, references, citings,

19 Concepts and implementation of a rule-based process engine
Burkhard Peuschel, Wilhelm Schäfer
June 1992 Proceedings of the 14th international conference on Software engineer
Full text available: pdf(1.77 MB) Additional Information: full citation, references, citings, index terms

20 Polynomial-time program transformations in deductive databases
Yatin P. Saraiya
April 1990 Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on
Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references,

We investigate the complexity of various optimization techniques for logic dat polynomial-time algorithms for restricted versions of common program transf relaxation of these restrictions leads to NP-hardness. To this end, we define t queries, and show that while the 2-containment problem is in P, the 3-contain

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7
Best 200 shown

21 Decidability and undecidability results for the termination problem of active
James Bailey, Guozhu Dong, Kotagiri Ramamohanarao
May 1998 Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART sympos
Full text available: pdf(1.28 MB) Additional Information: full citation, references, citings

22 A framework for testing safety and effective computability of extended data
Ravi Krishnamurthy, Raghu Ramakrishnan, Oded Shmueli
June 1988 Proceedings of the 1988 ACM SIGMOD international conference on M
Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references,

This paper presents a methodology for testing a general logic program contain predicates for safety and effective computability. Safety is the property that t finite. A related issues is whether the evaluation strategy can effectively com consider these problems under the assumption that queries are evaluated usi ...

23 Universal relation systems: Functional dependencies on cyclic database sc
Kent Laver, Alberto O. Mendelzon, Marc H. Graham
May 1983 Proceedings of the 1983 ACM SIGMOD international conference on M
Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, refere

We study how functional dependencies affect the cyclicity of a database schem functional dependencies make a cyclic database scheme behave like an acycli every pairwise-consistent database state that satisfies the fd's is join-consiste fd-acyclicity over a restricted class of database schemes. We then give a table case that leads to a ...

24 Coordinating rule-based software processes with ESP
Paolo Ciancarini
July 1993 ACM Transactions on Software Engineering and Methodology (TOSEM
Full text available: pdf(1.71 MB) Additional Information: full citation, abstract, references,

ESP is a language for modeling rule-based software processes that take place environment. It is based on PoliS, an abstract coordination model that relies o tuples a la Linda. PoliS extends Linda aiming at the specification and coordina (Extended Shared Prolog) combines the PoliS mechanisms to deal with concu logic-programming language ...

Keywords: concurrency, logic programming, multiuser programming environm process, software process modeling

25 Cactis: a self-adaptive, concurrent implementation of an object-oriented da
Scott E. Hudson, Roger King
September 1989 ACM Transactions on Database Systems (TODS), Volume 1
Full text available: pdf(2.65 MB) Additional Information: full citation, abstract, references, citing

Cactis is an object-oriented, multiuser DBMS developed at the University of C functionally-defined data and uses techniques based on attributed graphs to o functionally-defined data. The implementation is self-adaptive in that the phy algorithms dynamically change in order to reduce disk access. The system is are some number of computations that must be performed t ...

26 Implementation of logical query languages for databases
Jeffrey D. Ullman
September 1985 ACM Transactions on Database Systems (TODS), Volume 1
Full text available: pdf(2.66 MB) Additional Information: full citation, abstract, references, citin

We examine methods of implementing queries about relational databases in t in first-order logic as a collection of Horn clauses. Because queries may be de of query evaluation do not always work, and a variety of strategies have been queries. We express such query evaluation techniques as “capture rule and predicates. One ess ...

27 A framework for testing database applications

David Chays, Saikat Dan, Phyllis G. Frankl, Filippos I. Vokolos, Elaine J. Weber

August 2000 ACM SIGSOFT Software Engineering Notes , Proceedings of the Intern and Analysis, Volume 25 Issue 5

Full text available: pdf(557.89 KB) Additional Information: full citation, abstract, reference

Database systems play an important role in nearly every modern organization focused on how to test them. This paper discusses issues arising in testing da to testing database applications. In testing such applications, the state of the operation plays an important role, along with the user's input and the system with meaningful dat ...

Keywords: database, software testing, test data

28 Static analysis techniques for predicting the behavior of active database rul

Alexander Aiken, Joseph M. Hellerstein, Jennifer Widom

March 1995 ACM Transactions on Database Systems (TODS), Volume 20 Iss

Full text available: pdf(2.79 MB) Additional Information: full citation, abstract, references, citing

This article gives methods for statically analyzing sets of active database rule guaranteed to terminate, (2) guaranteed to produce a unique final database s unique stream of observable actions. If the analysis determines that one of th isolates the rules responsible for the problem and determines criteria that, if analysis methods are presented ...

Keywords: active database systems, confluence, database rule processing, sta

29 Testing implications of data dependencies

David Maier, Alberto O. Mendelzon, Yehoshua Sagiv

December 1979 ACM Transactions on Database Systems (TODS), Volume 4

Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references,

Presented is a computation method—the chase—for testing imp data dependencies. The chase operates on tableaux similar to those of Aho, S previous tableau computation methods as special cases. By interpreting table templates for relations, it is possible to test implication of join dependencies (functional dependenc ...

Keywords: chase, data dependencies, functional dependencies, join dependen databases, tableaux

30 QProber: A system for automatic classification of hidden-Web databases
Luis Gravano, Panagiotis G. Ipeirotis, Mehran Sahami
January 2003 ACM Transactions on Information Systems (TOIS), Volume 21
Full text available: pdf(3.62 MB) Additional Information: full citation, abstract, reference

The contents of many valuable Web-accessible databases are only available t invisible to traditional Web "crawlers." Recently, commercial Web sites have s Web-accessible databases into Yahoo!-like hierarchical classification schemes system that automates this classification process by using a small number of classifiers. QProber can use a variety of types of ...

Keywords: Database classification, Web databases, hidden Web

31 Probe, count, and classify: categorizing hidden web databases
Panagiotis G. Ipeirotis, Luis Gravano, Mehran Sahami
May 2001 ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD internatio Volume 30 Issue 2
Full text available: pdf(389.34 KB) Additional Information: full citation, abstract, reference

The contents of many valuable web-accessible databases are o interfaces and are hence invisible to traditional web “cra estimated the size of this “hidden web” to be 50(“crawlable” web is only an estimated two billion sites have started to manually organize web-accessible databa: classification schemes ...

32 On the decidability and axiomatization of query finiteness in deductive data
Michael Kifer
July 1998 Journal of the ACM (JACM), Volume 45 Issue 4
Full text available: pdf(323.85 KB) Additional Information: full citation, abstract, referen

A database query is finite if its result consists of a finite sets tuples. For queri the problem of determining finiteness is, in general, undecidable.In this pape stronger kind of finiteness, which applies to Horn queries whose function sym infinite relations with finiteness constraints (abbr., F ...

Keywords: axiomatization, computability, finite queries, finiteness constraints constraints, query processing

33 A form-based approach for database analysis and design
Joobin Choobineh, Michael V. Mannino, Veronica P. Tseng
February 1992 Communications of the ACM, Volume 35 Issue 2
Full text available: pdf(8.75 MB) Additional Information: full citation, references, index terms, review

Keywords: form processing, view definition, view integration

34 Discovery of multi-level rules and exceptions from a distributed database
Rónán Páircéir, Sally McClean, Bryan Scotney
August 2000 Proceedings of the sixth ACM SIGKDD international conference on Kn
Full text available: pdf(132.32 KB) Additional Information: full citation, references, i

Keywords: aggregates, distributed databases, exception discovery, multi-leve sufficient statistics

35 Safe query languages for constraint databases
Peter Z. Revesz
March 1998 ACM Transactions on Database Systems (TODS), Volume 23 Is
Full text available: pdf(295.75 KB) Additional Information: full citation, abstract, reference

In the database framework of Kanellakis et al. [1990] it was argued that cons constraint databases as input and give other constraint databases that use th output. This closed-form requirement has been difficult to realize in constrain negation symbol. This paper describes a general approach to restricting const subsets that contain only programs ...

36 Closures of database hypergraphs
Domenico Saccà
October 1985 Journal of the ACM (JACM), Volume 32 Issue 4
Full text available: pdf(2.51 MB) Additional Information: full citation, abstract, references, citi

A hypergraph formalism is introduced to represent database schemata. In par one full join dependency and a set of functional dependencies, is represented both undirected and directed hyperedges. Undirected hyperedges correspond directed hyperedges correspond to the functional dependencies. In addition, t

37 Rule based database access control—a practical approach
Tor Didriksen
November 1997 Proceedings of the second ACM workshop on Role-based access co
Full text available: pdf(1.01 MB) Additional Information: full citation, references, index terms

38 Rule-based optimization and query processing in an extensible geometric c
Ludger Becker, Ralf Hartmut Güting
June 1992 ACM Transactions on Database Systems (TODS), Volume 17 Issu
Full text available: pdf(3.35 MB) Additional Information: full citation, abstract, references, citin

Gral is an extensible database system, based on the formal concept of a man algebra is used to define any application's query language, its query executio this paper we describe Gral's optimization component. It provides (1) a sophi transformations of abstract algebra expressions, (2) a general optimization fra optimization algorithms can be ...

Keywords: extensibility, geometric query processing, many-sorted algebra, op optimization

39 Answering queries on embedded-complete database schemes
Edward P. F. Chan, Alberto O. Mendelzon
April 1987 Journal of the ACM (JACM), Volume 34 Issue 2
Full text available: pdf(2.22 MB) Additional Information: full citation, abstract, references, citin

It has been observed that, for some database schemes, users may have diffic for simple queries. The problem occurs when some implicit “piece&rdqu of a relation scheme, is not explicitly represented in the database state. In th how the state and the constraints interact before they can retrieve the inform not ...

40 Measuring system normality
Mark Burgess, Hårek Haugerud, Sigmund Straumsnes, Trond Reitan
May 2002 ACM Transactions on Computer Systems (TOCS), Volume 20 Issu
Full text available: pdf(794.43 KB) Additional Information: full citation, abstract, refere

A comparative analysis of transaction time-series is made, for light to modera problem of anomaly detection in computers. Criteria for measuring the statist a scaling transformation to the measured data, it is found that the distribution approximated by a steady-state, maximum-entropy distribution, modulated b distribution, under these con ...

Keywords: Anomaly detection, statistical mechanics

Results 21 - 40 of 200 Result page: previous 1 **2** 3

Results 41 - 60 of 200 Result page: previous 1 2 **3** 4 5 6 7
Best 200 shown

41 Special issue on prototypes of deductive database systems: The glue-nail (implementation, and evaluation
Marcia A. Derr, Shinichi Morishita, Geoffrey Phipps
April 1994 The VLDB Journal — The International Journal on Very Large D
Full text available: pdf(2.16 MB) Additional Information: full citation, abstract,

We describe the design and implementation of the Glue-Nail deductive databa query language; Glue is a procedural language used for non-query activities. to write a complete application. Nail and Glue code are both compiled into the uses variants of the magic sets algorithm and supports well-founded models. peephole techniques and data ...

Keywords: language, performance, query optimization

42 Exploratory mining and pruning optimizations of constrained associations r
Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, Alex Pang
June 1998 ACM SIGMOD Record , Proceedings of the 1998 ACM SIGMOD internati Volume 27 Issue 2
Full text available: pdf(1.65 MB) Additional Information: full citation, abstract, references,

From the standpoint of supporting human-centered discovery of knowledge, t rules suffers from the following serious shortcomings: (i) lack of user explora rigid notion of relationships. In effect, this model functions as a black-box, ad We propose, in this paper, an architecture that opens up the black-box, and s explorat ...

43 Fuzzy functional dependencies and lossless join decomposition of fuzzy rel
K. V. S. V. N. Raju, Arun K. Majumdar
June 1988 ACM Transactions on Database Systems (TODS), Volume 13 Issu
Full text available: pdf(3.05 MB) Additional Information: full citation, abstract, references, citin

This paper deals with the application of fuzzy logic in a relational database en more meaning of the data. It is shown that with suitable interpretations for th relational data model can be used to represent ambiguities in data values as among them. Relational operators for fuzzy relations have been studied, and integrity constraint ...

44 Automatic verification of database transaction safety
Tim Sheard, David Stemple
September 1989 ACM Transactions on Database Systems (TODS), Volume 1
Full text available: pdf(3.34 MB) Additional Information: full citation, abstract, references, citin

Maintaining the integrity of databases is one of the promises of database man that integrity constraints are invariants of database transactions. This is very presence of complex constraints and large amounts of data. One way to minim maintain database integrity over transaction processing is to prove at compile atomically, disobey i ...

45 Extending performance approaches to new application domains: An optimi configuration

David Bartholomew Stewart, Efstathios Papaefstathiou, Jonathan Hardwick

July 2002 Proceedings of the third international workshop on Software and per

Full text available: pdf(220.16 KB) Additional Information: full citation, abstrac

A common problem that sales consultants face in the field is the selection of a configuration for web farms. Over-provisioning means that the tender will be lead to a configuration that does not meet the customer criteria. Indy is a per allows developers to create custom modeling applications. We have construct web farm workloads and topologies. T ...

Keywords: design, experimentation, indy, infrastructures, measurement, mod simulation

46 A methodology for creating user views in database design

Veda C. Storey, Robert C. Goldstein

September 1988 ACM Transactions on Database Systems (TODS), Volume 1

Full text available: pdf(2.41 MB) Additional Information: full citation, abstract, references, citin

The View Creation System (VCS) is an expert system that engages a user in a requirements for some application, develops an Entity-Relationship model for converts the E-R model to a set of Fourth Normal Form relations. This paper d is, it presents a formal methodology, capable of mechanization as a computer a user, identifying and resolving incons ...

47 Optimization of a cycle time and utilization in semiconductor test manufacti near-real-time scheduling system

Appa Iyer Sivakumar

December 1999 Proceedings of the 31st conference on Winter simulation: Simulati

Full text available: pdf(142.30 KB) Additional Information: full citation, abstract, references,

48 Process synchronization in database systems

Gunter Schlageter

September 1978 ACM Transactions on Database Systems (TODS), Volume

Full text available: pdf(1.87 MB) Additional Information: full citation, abstract, references,

The problem of process synchronization in database systems is analyzed in a abstract level; the abstraction is chosen such that the essential characteristic and investigated. Using a small set of concepts, a consistent description of th used, but only vaguely defined, notions are defined exactly within this framew problem immediately leads ...

Keywords: database consistency, database systems, integrity, locking, paralle

49 Heraclitus: elevating deltas to be first-class citizens in a database program

Shahram Ghandeharizadeh, Richard Hull, Dean Jacobs

September 1996 ACM Transactions on Database Systems (TODS), Volume 2

Full text available: pdf(3.76 MB) Additional Information: full citation, abstract, references, citin

Traditional database systems provide a user with the ability to query and man current database state. However, in several emerging applications, the ability scenarios in order to reason about the impact of an update (before committin Example applications include hypothetical database access, active database m management, to name a few. The central th ...

Keywords: active databases, deltas, execution model for rule application, hyp state

50 An integrated general purpose automated test environment

Peter A. Vogel

July 1993 ACM SIGSOFT Software Engineering Notes , Proceedings of the 1993 in and analysis, Volume 18 Issue 3

Full text available: pdf(693.91 KB) Additional Information: full citation, abstract, reference

As software systems become more and more complex, both the complexity of maintaining the results of that effort increase proportionately. Most existing t flexibility needed to adequately test significant software systems. The CONVE discussed as an answer to the need for a more complete and powerful genera

51 A unified version model for configuration management
Andreas Zeller
October 1995 ACM SIGSOFT Software Engineering Notes , Proceedings of the 3rd A of software engineering, Volume 20 Issue 4
Full text available: pdf(1.02 MB) Additional Information: full citation, references, citin

52 A Theory of Safe Locking Policies in Database Systems
Mihalis Yannakakis
July 1982 Journal of the ACM (JACM), Volume 29 Issue 3
Full text available: pdf(1.33 MB) Additional Information: full citation, references, citings, index terms

53 Mining association rules between sets of items in large databases
Rakesh Agrawal, Tomasz Imieli?ski, Arun Swami
June 1993 ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD internati Volume 22 Issue 2
Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references,

We are given a large database of customer transactions. Each transaction con visit. We present an efficient algorithm that generates all significant associati The algorithm incorporates buffer management and novel estimation and pru applying this algorithm to sales data obtained from a large retailing company algorithm.

54 Integrating association rule mining with relational database systems: altern
Sunita Sarawagi, Shiby Thomas, Rakesh Agrawal
June 1998 ACM SIGMOD Record , Proceedings of the 1998 ACM SIGMOD internati Volume 27 Issue 2
Full text available: pdf(2.03 MB) Additional Information: full citation, abstract, references,

Data mining on large data warehouses is becoming increasingly important. In spectrum of architectural alternatives for coupling mining with database syste loose-coupling through a SQL cursor interface; encapsulation of a mining algo data to a file system on-the-fly and mining; tight-coupling using primarily use implementations for processing in the DBMS. We ...

55 Maintaining state constraints in relational databases: a proof theoretic basi
William W. McCune, Lawrence J. Henschen
January 1989 Journal of the ACM (JACM), Volume 36 Issue 1
Full text available: pdf(1.79 MB) Additional Information: full citation, abstract, references, citin

If a relational database is required to satisfy a set of integrity constraints, the ensure that it continues to satisfy the constraints. It is desirable not to have t update. A method is described that takes a constraint C and a class of update class cannot violate C, or produces a formula C' (a complete test) that is satis

56 Semantics of query languages for network databases
Kazimierz Subieta
September 1985 ACM Transactions on Database Systems (TODS), Volume 1
Full text available: pdf(3.71 MB) Additional Information: full citation, abstract, referenc

Semantics determines the meaning of language constructs; hence it says muc implementing the language. The main purpose of this paper is a formal prese constructs employed in many database languages (sublanguages). Therefore, Selection Language) and J (Joins) are introduced, wherein most of the typical are collected. The semantics of SSL and J are ...

57 A design rule database system to support technology-adaptable applicatior
J. S. Aude, Hillary J. Kahn
July 1986 Proceedings of the 23rd ACM/IEEE conference on Design automatio
Full text available: pdf(722.49 KB) Additional Information: full citation, abstract, references

This paper describes aspects of a CAD system which has been specifically des applications and to incorporate expert system processes where appropriate. T use design rules stored in a database to supply technology related information supported in the database are concerned with different aspects of design, suc rules are described i ...

58 On the Desirability of Acyclic Database Schemes
Catriel Beeri, Ronald Fagin, David Maier, Mihalis Yannakakis
July 1983 Journal of the ACM (JACM), Volume 30 Issue 3
Full text available: pdf(2.10 MB) Additional Information: full citation, references, citings, index terms

59 Design issues in a Rule-Based System
Stephen Fickas
June 1985 Proceedings of the ACM SIGPLAN 85 symposium on Language issues in
18 Issue 7 , 6
Full text available: pdf(733.76 KB) Additional Information: full citation, abstract, refer

This paper discusses a language and associated environment for building rule environment are encapsulated in a system we call ORBS (Oregon Rule Based focus will be on the interplay between language and environment design. How include design constraints placed by our program development model1 as we rationalization o ...

60 A database interface for file update
Serge Abiteboul, Sophie Cluet, Tova Milo
May 1995 ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD internatio
Volume 24 Issue 2
Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references,

Database systems are concerned with structured data. Unfortunately, data is manner (e.g., in files) even when it does have a strong internal structure (e.g previous paper [2], we focussed on the use of high-level query languages to a optimization techniques to do so. In this paper, we consider how structured d database update languages. ...

USPTO PATENT FULL-TEXT AND IMAGE DATABASE

Searching 1790 to present...

Results of Search in 1790 to present db for:
(((test AND "configuration engine") AND rule) AND database): 7 patents.
Hits 1 through 7 out of 7

Jump To

Refine Search `test and "configuration engine" and rule and databa`

	PAT. NO.	Title
1	6,430,730	Flash configuration cache
2	6,119,174	Methods and apparatus for implementing quality-of-service guarantees in data storage systems
3	6,115,547	Flash configuration cache
4	6,016,393	System and method for distributed computation based upon the movement, execution, and interaction of processes in a network
5	5,809,212	Conditional transition networks and computational processes for use interactive computer-based systems
6	5,603,031	System and method for distributed computation based upon the movement, execution, and interaction of processes in a network
7	5,452,239	Method of removing gated clocks from the clock nets of a netlist for timing sensitive implementation of the netlist in a hardware emulation system

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US	5458

33438 7590 12/30/2003

HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

EXAMINER	
STARKS, WILBERT L	
ART UNIT	**PAPER NUMBER**
2121	7

DATE MAILED: 12/30/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	09/773,101	GILPIN ET AL.	
	Examiner	Art Unit	
	Wilbert L. Starks, Jr.	2121	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 January 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-76* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-76* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

Claim Rejections - 35 USC § 101

1. 35 U.S.C. §101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

the invention as disclosed in claims 1-76 is directed to non-statutory subject matter.

2. Claims 1-76 are not claimed to be practiced on a computer, therefore, it is clear that the claims are not limited to practice in the technological arts. On that basis alone, they are clearly nonstatutory.

3. Regardless of whether any of the claims are in the technological arts, none of them is limited to practical applications in the technological arts. Examiner finds that *In re Warmerdam*, 33 F.3d 1354, 31 USPQ2d 1754 (Fed. Cir. 1994) controls the 35 USC §101 issues on that point for reasons made clear by the Federal Circuit in *AT&T Corp. v. Excel Communications, Inc.*, 50 USPQ2d 1447 (Fed. Cir. 1999). Specifically, the Federal Circuit held that the act of:

> ...[T]aking several abstract ideas and manipulating them together adds nothing to the basic equation. *AT&T v. Excel* at 1453 quoting *In re Warmerdam*, 33 F.3d 1354, 1360 (Fed. Cir. 1994).

Examiner finds that Applicant's "test case" references are just such abstract ideas.

4. Examiner bases his position upon guidance provided by the Federal Circuit in *In re Warmerdam*, as interpreted by *AT&T v. Excel*. This set of precedents is within the same line of cases as the *Alappat-State Street Bank* decisions and is in complete agreement with those decisions. *Warmerdam* is consistent with *State Street*'s holding that:

> Today we hold that *the transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price*, constitutes a practical application of a mathematical algorithm, formula, or calculation because it produces 'a useful, concrete and tangible result" – *a final share price momentarily fixed for recording purposes and even accepted and relied upon by regulatory authorities and in subsequent trades.* (emphasis added) *State Street Bank* at 1601.

5. True enough, that case later eliminated the "business method exception" in order to show that business methods were not per se nonstatutory, but the court clearly *did not* go so far as to make business methods *per se statutory*. A plain reading of the excerpt above shows that the Court was *very specific* in its definition of the new *practical application*. It would have been much easier for the court to say that "business methods were per se statutory" than it was to define the practical application in the case as "...the transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price..."

6. The court was being very specific.

7. Additionally, the court was also careful to specify that the "useful, concrete and tangible result" it found was "a final share price momentarily fixed for recording purposes and even accepted and relied upon by regulatory authorities and in subsequent trades." (i.e. the trading activity is the further practical use of the real world monetary data beyond the transformation in the computer – i.e., "post-processing activity".)

8. Applicant cites no such specific results to define a useful, concrete and tangible result. Neither does Applicant specify the associated practical application with the kind of specificity the Federal Circuit used.

9. Furthermore, in the case *In re Warmerdam*, the Federal Circuit held that:

> ...[T]he dispositive issue for assessing compliance with Section 101 in this case is whether the claim is for a process that goes beyond simply manipulating 'abstract ideas' or 'natural phenomena' ... As the Supreme Court has made clear, '[a]n idea of itself is not patentable, ... *taking several abstract ideas and manipulating them together adds nothing to the basic equation*. In re Warmerdam 31 USPQ2d at 1759 (emphasis added).

10. Since the Federal Circuit held in *Warmerdam* that this is the "dispositive issue" when it judged the usefulness, concreteness, and tangibility of the claim limitations in that case, Examiner in the present case views this holding as the dispositive issue for determining whether a claim is "useful, concrete, and tangible" in similar cases. Accordingly, the Examiner finds that Applicant manipulated a set of abstract "test cases" to solve purely algorithmic problems in the abstract (i.e., what *kind* of "test case" is used? Algebraic word problems? Boolean logic problems? Fuzzy logic algorithms? Probabilistic word problems? Philosophical ideas? Even vague expressions, about which even reasonable persons could differ as to their meaning? Combinations thereof?) Clearly, a claim for manipulation of "test cases" is provably even more abstract (and thereby less limited in practical application) than pure "mathematical algorithms" which the Supreme Court has held are per se nonstatutory – in fact, it *includes* the expression of nonstatutory mathematical algorithms.

11. Since the claims are not limited to exclude such abstractions, the broadest reasonable interpretation of the claim limitations includes such abstractions. Therefore, the claims are impermissibly abstract under 35 U.S.C. 101 doctrine.

12. Since *Warmerdam* is within the *Alappat-State Street Bank* line of cases, it takes the same view of "useful, concrete, and tangible" the Federal Circuit applied in *State Street Bank*. Therefore, under *State Street Bank*, this could not be a "useful, concrete and tangible result". There is only manipulation of abstract ideas.

13. The Federal Circuit validated the use of *Warmerdam* in its more recent *AT&T Corp. v. Excel Communications, Inc.* decision. The Court reminded us that:

> Finally, **the decision in In re Warmerdam**, 33 F.3d 1354, 31 USPQ2d 1754 (Fed. Cir. 1994) **is not to the contrary**. *** The court found that the claimed process did nothing more than manipulate basic mathematical constructs and concluded that **'taking several abstract ideas and manipulating them together adds nothing to the basic equation'**; hence, the court held that the claims were properly rejected under §101 ... Whether one agrees with the court's conclusion on the facts, **the holding of the case is a straightforward application of the basic principle that mere laws of nature, natural phenomena, and abstract ideas are not within the categories of inventions or discoveries that may be patented under §101**. (emphasis added) AT&T Corp. v. Excel Communications, Inc., 50 USPQ2d 1447, 1453 (Fed. Cir. 1999).

14. Remember that in *In re Warmerdam*, the Court said that this was the dispositive issue to be considered. In the *AT&T* decision cited above, the Court reaffirms that this is the issue for assessing the "useful, concrete, and tangible" nature of a set of claims under 101 doctrine. Accordingly, Examiner views the *Warmerdam* holding as the dispositive issue in this analogous case.

15. The fact that the invention is merely the manipulation of *abstract ideas* is clear. The data referred to by Applicant's phrase "test case" is simply an abstract construct that does not limit the claims to the transformation of real world data (such as monetary data or heart rhythm data) by some disclosed process. Consequently, the necessary conclusion under *AT&T, State Street* and *Warmerdam,* is straightforward and clear.

The claims take several abstract ideas (i.e., "test case" in the abstract) and manipulate them together adding nothing to the basic equation. Claims 1-76 are, thereby, rejected under 35 U.S.C. 101.

16. Regarding the "system" recitals in claims 27 – 45 and 70 – 76 and the presumed "product of manufacture" claims in claims 14 – 26, the invention is still found to be nonstatutory. Any other finding would be at variance with current case law. Specifically, the Federal Circuit held in *AT&T v. Excel*, 50 USPQ2d 1447 (Fed. Cir. 1999) that:

> Whether stated implicitly or explicitly, we consider the scope of Section 101 to be **the same regardless of the form** – machine or process – in which a particular claim is drafted. AT&T v. Excel, 50 USPQ2d 1447, 1452 citing In re Alappat, 33 F.3d at 1581, 31 USPQ2d at 1589 (Rader, J., concurring) (emphasis added.)

17. Examiner considers the scope of Section 101 to be the same regardless of whether Applicant *claims* a "process", "machine", or "product of manufacture". While the "system" recitals in the preambles of claims 27 – 45 and 70 – 76 make the claims ostensibly drawn to be "apparatus" claims, they are insufficient by themselves to limit the claims to statutory subject matter. Likewise, the presumed attempts to limit claims 14 – 26 to "product of manufacture" claims are insufficient by themselves to limit the claims to statutory subject matter. Examiner's position is clearly consistent with *Alappat,* and *AT&T* and is implicitly consistent with *Warmerdam* and *State Street.* Accordingly, those claims are also properly rejected.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-76 are rejected under 35 U.S.C. 112, first paragraph because current case law (and accordingly, the MPEP) require such a rejection if a 101 rejection is given because when Applicant has not in fact disclosed the practical application for the invention, as a matter of law there is no way Applicant could have disclosed *how* to practice the *undisclosed* practical application. This is how the MPEP puts it:

> ("The how to use prong of section 112 **incorporates as a matter of law** the requirement of 35 U.S.C. 101 that the specification disclose as a matter of fact a practical utility for the invention.... If the application fails as a matter of fact to satisfy 35 U.S.C. § 101, then the application **also fails as a matter of law** to enable one of ordinary skill in the art to use the invention under 35 U.S.C. § 112."); In re Kirk, 376 F.2d 936, 942, 153 USPQ 48, 53 (CCPA 1967) ("Necessarily, compliance with § 112 requires a description of how to use presently useful inventions, **otherwise an applicant would anomalously be required to teach how to use a useless invention**.").See, MPEP 2107.01(IV), quoting In re Kirk (emphasis added).

Therefore, claims 1—76 are rejected on this basis.

Conclusion

18. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

A. Shasha (U.S. Patent Number 5,809,212; dated 15 September 1998; class 706; subclass 046) discloses conditional transition networks.

B. White et al (U.S. Patent Number 5,603,031; dated 11 February 1997; class 709; subclass 317) discloses distributed computation.

C. Dai et al (U.S. Patent Number 5,542,239; dated 19 September 1995; class 703; subclass 019) discloses implementation of a netlist.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Wilbert L. Starks, Jr. whose telephone number is (703) 305-0027.

Alternatively, inquiries may be directed to the following:

S. P. E. Anil Khatri	**(703) 305-0282**
After-final (FAX)	**(703) 746-7238**
Official (FAX)	**(703) 746-7239**
Non-Official/Draft (FAX)	**(703) 746-7240**

WLS

13 December 2003

Wilbert L. Starks, Jr.
Primary Examiner
Art Unit - 2121

Notice of References Cited	Application/Control No. 09/773,101	Applicant(s)/Patent Under Reexamination GILPIN ET AL.	
	Examiner Wilbert L. Starks, Jr.	Art Unit 2121	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,809,212	09-1998	Shasha, Dennis	706/46
	B	US-5,603,031	02-1997	White et al.	709/317
	C	US-5,452,239	09-1995	Dai et al.	703/19
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

2121

41

HAMILTON & TERRILE, LLP

8911 North Capital of Texas Highway
Westech Center Suite 3150
Austin, Texas 78759
512.338.9100 Telephone
512.345.7225 Facsimile

June 29, 2004

Mail Stop Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re:	Applicant(s):	Kevin E. Gilpin, et al.		
	Assignee:	Trilogy Development Group, Inc.		
	Title:	Rule Based Configuration Engine for a Database		
	Serial No.:	09/773,101	Filed:	January 31, 2001
	Examiner:	Wilbert L. Starks	Group Art Unit:	2121
	Docket No.:	T00011	Customer No.:	33438

Dear Sir:

Transmitted herewith are the following documents in the above-identified application:

(1) This Transmittal Letter;
(2) Petition for Extension of Time; and
(3) Response to Non-Final Office Action (23 pages).

☐ No additional fee is required.
☒ The fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining After Amendment		Highest No. Previously Paid For		Present Extra		Rate	Additional Fee
Total Claims	71	Minus	76	=	0	x	$18	$ .00
Independent Claims	4	Minus	8	=	0	x	$86	$

☒ Fee for Request for Extension of Time (3 months) $ 950.00

☒ Check Enclosed for Total Fee for this Amendment: $ 950.00

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account 502264.

I hereby certify that this is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Fee Amendment, COMMISSIONER FOR PATENTS, P.O. Box 1450, Alexandria, VA 22313-1450, on June 29, 2004.

Attorney for Applicant(s) — Date of Signature: 6-29-2004

Respectfully submitted,

Kent B. Chambers
Attorney for Applicant(s)
Reg. No. 38,839

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Kevin E. Gilpin, et al.		
Assignee:	Trilogy Development Group		
Title:	RULE BASED CONFIGURATION ENGINE FOR A DATABASE		
Serial No.:	09/773,101	Filed:	January 31, 2001
Examiner:	Wilbert L. Starks	Group Art Unit:	2121
Docket No.:	T00011	Customer No.:	33438

Austin, Texas
June 29, 2004

MAIL STOP FEE AMENDMENT
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450

RESPONSE TO NON-FINAL OFFICE ACTION

Dear Sir:

This paper is responsive to the Office action dated December 30, 2003, having a shortened statutory period expiring March 30, 2004. Accompanying this response is a petition under 37 C.F.R. § 1.136 for extension of time by three (3) months, setting a new time for response of June 30, 2004. Further examination and reconsideration are respectfully requested in view of the amendments and remarks set forth below.

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of ~~testing~~ using a computer system to test a product configuration for configuration errors, wherein the product configuration is stored as electronic data in a computer system for generating product configurations, the computer system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the method comprising:

entering a test case into the computer system to detect configuration errors in the product configuration, wherein the test case includes data to ~~selects at least one part to include in~~ change the product configuration; ~~and~~

processing the test case with the computer system in accordance with the at least one rule to ~~determine~~ detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error ~~at least one part selected in the test case conflicts with the plurality of parts previously included~~ in the product configuration; and

generating explanation data with the computer system to provide an explanation of any detected configuration error in the product configuration.

2. (Currently amended) The method, as set forth in claim 1, wherein processing the ~~at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration~~ test case, further includes:

initializing the computer system with a part state;

inputting ~~the~~ at least one part selection to change the product configuration; and

listening to state change events in the system to detect when a state change event occurs that results in the computer system being in the initialized part state.

3. (Currently amended) The method, as set forth in claim 2, wherein ~~processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration~~ generating explanation data, further includes:

generating ~~a cause~~ explanation data that explains the part state in terms of the state change event.

4. (Currently amended) The method, as set forth in claim 3, wherein processing the ~~at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration~~ test case, further includes:

generating a new part state for each part associated with the ~~cause~~ change in the product configuration.

5. (Currently amended) The method, as set forth in claim 4~~,~~ wherein processing the ~~at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration,~~ test case further includes:

determining ~~the~~ causes that explain the new part states in terms of the state change event.

6. (Currently amended) The method, as set forth in claim 5~~,~~ wherein generating explanation data further ~~comprising~~comprises:

generating a cause tree wherein the root of the cause tree is the initial part state~~,~~ and leaves of the tree are the user's selections of parts.

7. (Currently amended) The method, as set forth in claim 6 wherein generating explanation data comprises~~,~~ further ~~comprising~~ comprises:

generating an explanation of the part state wherein the part selections are the root of the explanation data and the causes follow from the part selections.

8. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the explanation data is based on selection of a part.

9. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the explanation data is based on execution of a rule.

10. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the explanation data is based on a part being in two states at the same time.

11. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the explanation data is based on a requires choice rule that cannot be satisfied.

12. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the explanation data is based on a look ahead process.

13. (Original) The method, as set forth in claim 7, further comprising:

sorting the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

14. (Currently amended) ~~An article of manufacture~~ A computer program product having code embodiment therein to cause a processor to test a product configuration for configuration errors, wherein the product configuration is stored as electronic data in a computer system, the computer system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the code comprising:

~~a computer usable medium having computer readable program code embodied therein for testing a product configuration in a system for generating product configurations, the system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the computer readable program code including:~~

computer readable program code configured to cause ~~a~~ the computer system to allow a user to enter a test case into the computer system to detect configuration errors in the product configuration, wherein the test case

includes data to ~~selects at least one part to include in~~ change the product configuration;~~ and~~

computer readable program code configured to cause ~~a~~ the computer system to process the test case with the computer system in accordance with the at least one rule to ~~determine~~ detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error~~at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.~~; and

computer readable program code configured to cause the computer system to generate explanation data with the computer system to provide an explanation of any detected configuration error in the product configuration.

15. (Currently amended) The article of manufacture, as set forth in claim 14, further including:

computer readable program code configured to cause ~~a~~ the computer system to initialize the computer system with a part state;

computer readable program code configured to cause ~~a~~ the computer system to input ~~the~~ at least one part selection to change the product configuration; and

computer readable program code configured to cause ~~a~~ the computer system to listen to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

16. (Currently amended) The article of manufacture, as set forth in claim 15, further including:

computer readable program code configured to cause ~~a computer~~ the computer system to generate ~~a cause~~ explanation data that explains the part state in terms of the state change event.

17. (Currently amended) The article of manufacture, as set forth in claim 16, further including:

computer readable program code configured to cause ~~a computer~~ the computer system to generate a new part state for each part associated with the ~~cause~~ change in the product configuration.

18. (Currently amended) The article of manufacture, as set forth in claim 17, further including:

computer readable program code configured to cause ~~a computer~~ the computer system to determine ~~the~~ causes that explain the new part states in terms of the state change event.

19. (Currently amended) The article of manufacture, as set forth in claim 18, further comprising:

computer readable program code configured to cause ~~a computer~~ the computer system to generate a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

20. (Currently amended) The article of manufacture, as set forth in claim 19, further comprising:

computer readable program code configured to cause ~~a computer~~ the computer system to generate an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

21. (Currently amended) The article of manufacture, as set forth in claim ~~20~~14, wherein the explanation data is based on selection of a part.

22. (Currently amended) The article of manufacture, as set forth in claim ~~20~~14, wherein the explanation data is based on execution of a rule.

23. (Currently amended) The article of manufacture, as set forth in claim ~~20~~14, wherein the explanation data is based on a part being in two states at the same time.

24. (Currently amended) The article of manufacture, as set forth in claim ~~20~~14, wherein the explanation data is based on a requires a choice rule that cannot be satisfied.

25. (Currently amended) The article of manufacture, as set forth in claim ~~20~~14, wherein the explanation data is based on a look ahead process.

26. (Currently amended) The article of manufacture, as set forth in claim 20, further comprising:

computer readable program code configured to cause ~~a computer~~ the computer system to sort the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

27. (Currently amended) An apparatus for testing a product configuration for configuration errors generated by a product configuration system, comprising:

a memory having stored therein at least one rule defining a relationship between at least two parts in the product configuration;

a test case to detect configuration errors in the product configuration, wherein the test case includes data to change the product configuration ~~pertaining to at least one part to include in the product configuration~~; and

a processor coupled to the memory to (a) process ~~receive~~ the at least one rule and the test case, ~~wherein the processor is operable to determine~~ (b) detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error ~~at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to the at least one rule~~ and (c) generate explanation data to provide an explanation of any detected configuration error in the product configuration.

28. (Currently amended) The apparatus, as set forth in claim 27, wherein the processor is further operable to:

initialize the configuration system with a part state;

to input the at least one part selection to change the product configuration;

to listen to state change events in the system; and

to detect when a state change event occurs that results in the configuration system being in the initialized part state.

29. (Currently amended) The apparatus, as set forth in claim 28, wherein the processor is further operable to:

generate ~~a cause~~ explanation data that explains the part state in terms of the state change event.

30. (Currently amended) The apparatus, as set forth in claim 29, wherein the processor is further operable to:

generate a new part state for each part associated with the ~~cause~~ change in the product configuration.

31. (Original) The apparatus, as set forth in claim 30, wherein the processor is further operable to:

generate a cause tree wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

32. (Original) The apparatus, as set forth in claim 30, wherein the processor is further operable to:

generate an explanation of the part state wherein the part selections are the root of the explanation and the causes follow from the part selections.

33. (Currently amended) The apparatus, as set forth in claim ~~32~~27, wherein the explanation data is based on execution of a rule.

34. (Currently amended) The apparatus, as set forth in claim ~~32~~ 27, wherein the explanation data is based on a part being in two states at the same time.

35. (Currently amended) The apparatus, as set forth in claim ~~32~~ 27, wherein the explanation data is based on a requires a choice rule that cannot be satisfied.

36. (Currently amended) The apparatus, as set forth in claim ~~32~~ 27, wherein the explanation data is based on a look ahead process.

37. (Original) The apparatus, as set forth in claim 30, wherein the processor is further operable to:

sort the tree by iteration number, wherein the iteration number of a part state is determined by measuring the longest distance between the part state and the cause corresponding to the part state.

38. (Canceled).

39. (Currently amended) The ~~configuration system of claim 38~~ apparatus as set forth in claim 27, wherein the test case further includes ~~the~~ a product selection.

40. (Currently amended) The ~~configuration system of claim 38~~ apparatus as set forth in claim 27 wherein the product configuration comprises~~, further comprising:~~ at least one vector, wherein said vector comprises a bit field, further wherein the bit field comprises bits that represent elements in a configuration.

41. (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in claim 40, wherein the number of bits in the bit field is equal to the total number of elements and an element's bit can be set or reset to specify that state of the element in the configuration.

42. (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in claim 40, wherein the vector specifies whether an element has been selected by the user during the configuration.

43. (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in claim 40, wherein excluded vectors identify whether an element is excluded from a configuration.

44. (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in claim 40, wherein removed vectors identify whether an element is removed from a configuration.

45. (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in claim 40, wherein the vector identifies whether an element is selectable.

46. (Currently Amended) ~~A database~~The apparatus as set forth in claim 40 further comprising:

a database having at least one table, wherein said table represents relationships between elements in a configuration~~;~~ and having at ~~lease~~ least one modified rule, wherein the rule is modified based on ~~the~~ results of testing a product selection.

47. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46, wherein said table represents "includes" relationships between elements in a configuration.

48. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46, wherein said table represents "excludes" relationships between elements in a configuration.

49. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46, wherein said table represents "removes" relationships between elements in a configuration.

50. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46, wherein said table represents "requires choice" relationships between elements in a configuration.

51. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 50, wherein the representation of "requires choice" relationships includes a pointer to a group table that includes a bit vector that identifies the elements that are contained in the group from which a choice is to be made.

52. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 50, wherein the representation of "requires choice" relationships includes minimum and maximum designations to identify the minimum and maximum number of group members that are to be selected to satisfy the "requires choice" relationship.

53. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46, wherein said table includes a left-hand side and a right-hand side.

54. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 53, wherein the left-hand side includes a bit vector that contains bits corresponding to elements.

55. (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 53, wherein the right-hand side includes one or more bit vectors that represent configuration elements.

56. (Currently amended) The apparatus as set forth in claim 27 wherein the test case further comprises data representing ~~A test case for testing a product configuration generated by a product configuration system, comprising~~:

a product selection;

at least one part selection; and

an expected state of the selected part based on one or more rules.

57. (Currently amended) ~~A method for identifying an invalid configuration generated by a product configuration system, comprising:~~ The method as set forth in claim 1 wherein the test case further comprises data to:

~~selecting~~ select a product;

~~selecting~~ select at least one part; and

~~generating~~ generate a part state of the selected part based on one or more rules.

58. (Canceled).

59. (Canceled).

60. (Canceled).

61. (Original) The method as set forth in claim 57, further comprising:
determining whether the product is selectable.

62. (Canceled).

63. (Original) The method as set forth in claim 57, further comprising:
reporting the state of the product as not selectable when selection of the product would conflict with the rule.

64. (Original) The method as set forth in claim 57, further comprising:
determining sets of parts that are excluded or deleted based on the product.

65. (Currently amended) The method as set forth in claim 57, further comprising:
detecting when a state change event occurs that results in the computer system being in the initialized part state.

66. (Canceled).

67. (Canceled).

68. (Canceled).

69. (Canceled).

70. (Currently amended) An apparatus for testing a product configuration for configuration errors generated by a computer implemented product configuration system, comprising:

means for defining a relationship between at least two parts in the product configuration;

means for defining a test case ~~for at least one part to include in the product configuration~~ to detect configuration errors in the product configuration, wherein the test case includes data to change the product configuration; ~~and~~

~~means for determining whether the at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to at least one rule~~

means for processing the test case with the product configuration system in accordance with the at least one rule to detect whether the change in the product configuration, as a result of processing the test case in accordance with the relationship between at least two parts in the product configuration, produced a configuration error in the product configuration; and

means for generating explanation data with the product configuration system to provide an explanation of any detected configuration error in the product configuration.

71. (Currently amended) The apparatus, as set forth in claim 70, further comprising:

means for initializing the configuration system with a part state;

means for detecting a state change event in the configuration system; and

means for detecting when a state change event occurs that results in the configuration system being in the initialized part state.

72. (Original) The apparatus, as set forth in claim 71, further comprising:

means for generating a cause that explains the part state in terms of the state change event.

73. (Original) The apparatus, as set forth in claim 72, further comprising:
means for generating a new part state for each part associated with the cause.

74. (Original) The apparatus, as set forth in claim 73, further comprising:
means for generating a cause tree, wherein the root of the cause tree is the initial part state, and leaves of the tree are the user's selections of parts.

75. (Original) The apparatus, as set forth in claim 73, further comprising:
means for generating an explanation of the part state, wherein the part selections are the root of the explanation and the causes follow from the part selections.

76. (Original) The apparatus, as set forth in claim 70, further comprising:
means for modifying the at least one rule when the test case conflicts with the plurality of parts previously included in the product configuration.

77. (New) The method, as set forth in claim 1, wherein the test case further includes data to select at least one part to include in the product configuration and processing test case further comprises:
processing the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

78. (New) The computer program product, as set forth in claim 14, wherein the test case further includes data to select at least one part to include in the product configuration and the computer readable program code configured to cause the computer system to process the test case further comprises:
computer readable code to process the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

79. (New) The apparatus, as set forth in claim 27, wherein the test case further pertains to including at least one part in the product configuration and the processor is further operable to:

determine whether the at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to the at least one rule.

80. (New) The apparatus, as set forth in claim 70, wherein the test case is further defined to include at least one part in the product configuration and the means for processing the test case includes:

means for determining whether the at least one part in the test case conflicts with the plurality of parts previously included in the product configuration according to the at least one rule.

REMARKS

Claims 1-76 are pending.

Claims 1-76 stand rejected.

Claims 1-12, 14-30, 33-36, 39-57, 65, and 70-71 have been amended.

Claims 38, 58-60, 62, and 66-69 have been cancelled without prejudice or disclaimer of the subject matter recited therein.

Claims 77-80 have been added.

Claim Rejections - 35 U.S.C. § 101

Claims 1-76 stand rejected under 35 U.S.C. § 101 as directed to non-statutory subject matter. The Office Action on p. 2, para. 2 states that "Claims 1-76 are not claimed to be practiced on a computer, therefore, it is clear that the claims are not limited to practice in the technological arts. On that basis alone they are clearly non-statutory." The Office Action on p. 2, para. 3 states that "Regardless of whether any of the claims are in the technological arts, none of them is limited to practical applications in the technological arts."

In light of the amendments to the claims and the remarks set forth herein, Applicants respectfully traverse the rejection.

The Federal Circuit in *AT&T* affirmed that "A mathematical formula alone, sometimes referred to as a mathematical algorithm, viewed in the abstract, is considered unpatentable subject matter." *AT&T v. Excel Communications, Inc.*, 50 U.S.P.Q.2d 1447 (Fed. Cir. 1999) (emphasis added). The Federal Circuit explained that the Supreme Court "never intended to create an overly broad, fourth category of [mathematical] subject matter excluded from § 101." *In re Alappat*, 31 USPQ2d 1545 (Fed. Cir. 1994). "Rather, at the core of the Court's analysis . . . lies an attempt by the Court to explain a rather straightforward concept, namely, that certain types of mathematical subject matter, standing alone , represent nothing more than abstract ideas until reduced to some type of practical application, and thus that subject matter is not, in and of

itself, entitled to patent protection." *Id.*, 31 USPQ2d 1545 (Fed. Cir. 1994) (emphasis added). "Thus, the *Alappat* inquiry simply requires an examination of the contested claims to see if the claimed subject matter as a whole is a disembodied mathematical concept representing nothing more than a "law of nature" or an "abstract idea," or if the mathematical concept has been reduced to some practical application rendering it "useful." *AT&T,* 50 U.S.P.Q.2d 1447 (Fed. Cir. 1999) (emphasis added). For example, in *AT&T* the Federal Circuit cited *State Street* as an example of a "claimed data processing system for implementing a financial management structure [that] satisfied the § 101 inquiry because it constituted a "practical application of a mathematical algorithm, . . . [by] produc[ing] `a useful, concrete and tangible result." *AT&T,* 50 U.S.P.Q.2d 1447 (Fed. Cir. 1999), *citing, State Street Bank & Trust Co. v. Signature Fin. Group, Inc.* , 47 USPQ2d 1596, 1602 (Fed. Cir. 1998), *cert. denied,* 119 S. Ct. 851 (1999).

The Examiner rejected claims 1-76 under 35 U.S.C. § 101 because the claims are "not limited to practice in the technological arts" and "none of them is limited to practical applications in the technological arts." Office Action, p. 2, paras. 2-3. More specifically, the Examiner "finds that Applicant's "test case" references are just such abstract ideas." *Id.*, para. 3. The Examiner also stated that "Applicant cites no such specific results to define a useful, concrete and tangible result." *Id.*, para. 7. "Neither does Applicant specify the associated practical application with the kind of specificity the Federal Circuit used." *Id.* The Examiner further stated that "the Examiner finds that Applicant manipulated a set of abstract "test cases" to solve purely algorithmic problems in the abstract (i.e. what *kind* of "test case" is used)?" The Examiner also stated that "Since the claims are not limited to exclude such abstractions, the broadest reasonable interpretation of the claim limitations includes such abstractions." *Id.*, para. 11. "Therefore, the claims are impermissibly abstract under 35 U.S.C. § 101 doctrine." *Id.*

As explained below, Applicants respectfully submit that the claims of the present application meet the statutory requirements of 35 U.S.C. § 101. Applying Federal Circuit law to the subject matter of the claims of the present application, to determine if the claims are non-statutory under 35 U.S.C. § 101 first "requires an examination of the contested claims to see if the claimed subject matter as a whole is a disembodied mathematical concept representing nothing more than a "law of nature" or an "abstract idea," or if the mathematical concept has been reduced to some practical application rendering it "useful." *AT&T,* 50 U.S.P.Q.2d 1447

(Fed. Cir. 1999). The Examiner has asserted that "Applicant's "test case" references are just such abstract ideas." Initially, Applicants respectfully submit that "test cases" are not merely abstract ideas. The Examiner asked "what *kind* of "test case" is used?" prior to stating that "Since the claims are not limited to exclude such abstractions, the broadest reasonable interpretation of the claim limitations includes such abstractions." Office action, paras. 10-11. This appears to be a question of scope outside the sphere of 35 U.S.C. § 101. The specification provides illustrative support for the term "test case", and the independent claims recite a specific type of test case, i.e. "a test case ... to detect configuration errors in the product configuration." Furthermore, the recited test case "includes data to change the product configuration." Claims 1, 14, 27, and 70. (Note: the present invention is limited by the claims and not by specific embodiments set forth in the description). Thus, the "test case" is not an abstract idea.

Even assuming *arguendo* that "test case" is an abstract idea, under Federal Circuit law that does not make a claim *per se* non-statutory under 35 U.S.C. § 101. The correct inquiry is whether the "claimed subject matter as a whole is a disembodied mathematical concept representing nothing more than a "law of nature" or an "abstract idea," or if the mathematical concept has been reduced to some practical application rendering it "useful." *AT&T,* 50 U.S.P.Q.2d 1447 (Fed. Cir. 1999) (emphasis added). The Supreme Court in *Diamond v. Diehr* explicitly distinguished Diehr's process by pointing out that "the respondents here do not seek to patent a mathematical formula. Instead, they seek patent protection for a process of curing synthetic rubber." *Diamond v. Diehr,* 450 U.S.175, 187 (1981). "The Court then explained that although the process used a well- known mathematical equation, the applicants did not "pre-empt the use of that equation." *AT&T,* 50 U.S.P.Q.2d 1447 (Fed. Cir. 1999), *citing Diehr*, 45 U.S. at 187. "Thus, even though a mathematical algorithm is not patentable in isolation, a process that applies an equation to a new and useful end "is at the very least not barred at the threshold by § 101." *AT&T,* 50 U.S.P.Q.2d 1447 (Fed. Cir. 1999), *citing Diehr*, 45 U.S. at 188. Likewise, the claims of the present application do not seek to patent a "test case" in isolation, i.e. in the abstract, and, thus, do not claim "a disembodied mathematical concept." *AT&T,* 50 U.S.P.Q.2d 1447 (Fed. Cir. 1999). To the contrary, rather than claiming a test case in the abstract, independent claims 1, 14, 27, and 70 recite a specific test case in the context of processes (claim 1) and components (claims 14, 27, and 70) to "test a product configuration for configuration

errors," "detect a configuration error", and "generat[e] explanation data." Specifically, claim 1 recites:

> A method of using a computer system to test a product configuration for configuration errors, wherein the product configuration is stored as electronic data in a computer system for generating product configurations, the computer system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the method comprising:
>
> entering a test case into the computer system to detect configuration errors in the product configuration, wherein the test case includes data to change the product configuration;
>
> processing the test case with the computer system in accordance with the at least one rule to detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error in the product configuration; and
>
> generating explanation data with the computer system to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Claim 14 recites:

> A computer program product having code embodiment therein to cause a processor to test a product configuration for configuration errors, wherein the product configuration is stored as electronic data in a computer system, the computer system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the code comprising:
>
> computer readable program code configured to cause the computer system to allow a user to enter a test case into the computer system to detect configuration errors in the product configuration, wherein the test case includes data to change the product configuration;
>
> computer readable program code configured to cause the computer system to process the test case with the computer system in accordance with the at least one rule to detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error; and
>
> computer readable program code configured to cause the computer system to generate explanation data with the computer system to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Claim 27 recites:

An apparatus for testing a product configuration for configuration errors generated by a product configuration system, comprising:

a memory having stored therein at least one rule defining a relationship between at least two parts in the product configuration;

a test case to detect configuration errors in the product configuration, wherein the test case includes data to change the product configuration; and

a processor coupled to the memory to (a) process the at least one rule and the test case, (b) detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error and (c) generate explanation data to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Claim 70 recites:

An apparatus for testing a product configuration for configuration errors generated by a computer implemented product configuration system, comprising:

means for defining a relationship between at least two parts in the product configuration;

means for defining a test case to detect configuration errors in the product configuration, wherein the test case includes data to change the product configuration; and

means for processing the test case with the product configuration system in accordance with the at least one rule to detect whether the change in the product configuration, as a result of processing the test case in accordance with the relationship between at least two parts in the product configuration, produced a configuration error in the product configuration; and

means for generating explanation data with the product configuration system to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Furthermore, each of independent claims provide "a new and useful end" and, thus, are not barred by 35 U.S.C. § 101. *quoting Diehr*, 45 U.S. at 188. Claim 1 recites specific processes that produce a new, useful, concrete, and tangible end:

processing the test case with the computer system in accordance with the at least one rule to detect whether the change in the product configuration ... produced a configuration error in the product configuration; and

generating explanation data with the computer system to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Claim 14 recites specific computer readable program code that produces a new, useful, concrete, and tangible end:

> ... to process the test case with the computer system in accordance with the at least one rule to detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error; and
> ... to cause the computer system to generate explanation data with the computer system to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Claim 27 recites specific components that produce a new, useful, concrete, and tangible end:

> ... a processor coupled to the memory to (a) process the at least one rule and the test case, (b) detect whether the change in the product configuration ... produced a configuration error and (c) generate explanation data to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Claim 70 also recites specific components that produce a new, useful, concrete, and tangible end:

> means for processing the test case with the product configuration system in accordance with the at least one rule to detect whether the change in the product configuration ... produced a configuration error in the product configuration; and
> means for generating explanation data with the product configuration system to provide an explanation of any detected configuration error in the product configuration. (emphasis added).

Thus, the claims of the present invention "test a product configuration for configuration errors" using processes and components that provide a new and useful end and have practical application, which conforms with the statutory requirements of 35 U.S.C. § 101 as supported by Federal Circuit case law.

Applicants respectfully submit that claims dependent upon independent claims 1, 14, 27, or 70, directly or indirectly, meet the statutory requirements of 35 U.S.C. § 101 for at least the same reasons as the independent claim upon which each depends.

Accordingly, withdrawal of the 35 U.S.C. § 101 rejection is respectfully requested.

Claim Rejections - 35 U.S.C. § 112

"Claims 1-76 are rejected under 35 U.S.C. § 112, first paragraph because current case law (and accordingly, the MPEP) require such a rejection if a 101 rejection is given." Office action, p. 8.

Since the basis for the 35 U.S.C. § 112, first paragraph rejection is essentially the same as the rejection under 35 U.S.C. § 101, Applicants respectfully request withdrawal of the 35 U.S.C. § 112, first paragraph rejection for at least the same reasons as those presented pursuant to the 35 U.S.C. § 101 rejection.

CONCLUSION

In view of the amendments and remarks set forth herein, the application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the examiner is requested to telephone the undersigned.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Fee Amendment, COMMISSIONER FOR PATENTS, P.O. Box 1450, Arlington, VA 22313-1450, on June 29, 2004.

Attorney for Applicant(s) June 29, 2004 Date of Signature

Respectfully submitted,

Kent B. Chambers
Attorney for Applicant(s)
Reg. No. 38,839

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Kevin E. Gilpin, et al.		
Assignee:	Trilogy Development Group, Inc.		
Title:	RULE BASED CONFIGURATION ENGINE FOR A DATABASE		
Serial No.:	09/773,101	Filed:	January 31, 2001
Examiner:	Wilbert L. Starks	Group Art Unit:	2121
Docket No.:	T00011	Customer No.:	33438

Austin, Texas
June 29, 2004

MAIL STOP FEE AMENDMENT
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR EXTENSION OF TIME

Dear Sir:

Applicants respectfully petition for a three (3) month extension of time within which to respond to the December 30, 2003, outstanding Office Action, such extension allowing the undersigned until June 30, 2004, to respond.

A check in the amount of $950 is enclosed to cover the fee for the requested extension of time. The Commissioner is authorized to deduct any additional fees which may be required or credit any overpayment to Deposit Account No. 502264.

I hereby certify that this is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Fee Amendment, COMMISSIONER FOR PATENTS, P.O. Box 1450, Alexandria, VA 22313-1450, on June 29, 2004.

Attorney for Applicant(s) — Date of Signature: June 29, 2004

Respectfully submitted,

Kent B. Chambers
Attorney for Applicant(s)
Reg. No. 38,839

07/06/2004 CNGUYEN 00000066 09773101
01 FC:1253 950.00 OP

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

Application or Docket Number: M-7822 US 09/723101

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	76	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	76 minus 20=	* 56
INDEPENDENT CLAIMS	8 minus 3 =	* 5
MULTIPLE DEPENDENT CLAIM PRESENT		☐

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

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	1,008
X40=		OR	X80=	400
+135=		OR	+270=	
TOTAL		OR	TOTAL	2118

CLAIMS AS AMENDED - PART II

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 71	Minus	** 76	= 0
Independent	* 4	Minus	*** 8	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

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

FORM PTO-875
(Rev. 8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-460-706/30103

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

33438 7590 08/13/2004

HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

EXAMINER	
STARKS, WILBERT L	
ART UNIT	PAPER NUMBER
2121	

DATE MAILED: 08/13/2004

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US	5458

TITLE OF INVENTION: RULE BASED CONFIGURATION ENGINE FOR A DATABASE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1330	$0	$1330	11/15/2004

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE REFLECTS A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE APPLIED IN THIS APPLICATION. THE PTOL-85B (OR AN EQUIVALENT) MUST BE RETURNED WITHIN THIS PERIOD EVEN IF NO FEE IS DUE OR THE APPLICATION WILL BE REGARDED AS ABANDONED.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL should be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). Even if the fee(s) have already been paid, Part B - Fee(s) Transmittal should be completed and returned. If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 07/04) Approved for use through 04/30/2007.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (703) 746-4000

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

33438 7590 08/13/2004

HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (703) 746-4000, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US	5458

TITLE OF INVENTION: RULE BASED CONFIGURATION ENGINE FOR A DATABASE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1330	$0	$1330	11/15/2004

EXAMINER	ART UNIT	CLASS-SUBCLASS
STARKS, WILBERT L	2121	706-001000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____

2 _____

3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent); ❑ individual ❑ corporation or other private group entity ❑ government

4a. The following fee(s) are enclosed:

❑ Issue Fee

❑ Publication Fee (No small entity discount permitted)

❑ Advance Order - # of Copies _____

4b. Payment of Fee(s):

❑ A check in the amount of the fee(s) is enclosed.

❑ Payment by credit card. Form PTO-2038 is attached.

❑ The Director is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

❑ b. Applicant is not claiming SMALL ENTITY status. See, e.g., 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

(Authorized Signature) (Date)

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMIT THIS FORM WITH FEE(S)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US	5458

33438 7590 08/13/2004

HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

EXAMINER	
STARKS, WILBERT L	

ART UNIT	PAPER NUMBER
2121	

DATE MAILED: 08/13/2004

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 546 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 546 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (703) 305-1383. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (703) 305-8283.

PTOL-85 (Rev. 07/04) Approved for use through 04/30/2007.

Notice of Allowability	Application No. 09/773,101	Applicant(s) GILPIN ET AL.	
	Examiner Wilbert L. Starks, Jr.	Art Unit 2121	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the amendment filed 01 July 2004*.

2. ☒ The allowed claim(s) is/are *1-37,39-57,61,63-65 and 70-80*.

3. ☒ The drawings filed on *01 July 2004* are accepted by the Examiner.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

Wilbert L. Starks, Jr.
Primary Examiner
Art Unit: 2121

DETAILED ACTION

Reasons For Allowance

1. Claims 1-37, 39-57, 61, 63-65, and 70-80 are allowed.

2. The following is an Examiner's statement of reasons for allowance:

The cited prior art taken alone or in combination fails to teach the claimed invention of a rule based configuration engine, as claimed by Applicant. Specifically, independent claims 1, 14, 27, and 70 disclose the use of a computer system to test an electronically stored product configuration for errors.

The closest prior art of Dai et al (U.S. Patent Number 5,542,239; dated 19 September 1995; class 703; subclass 019) teaches the implementation of a netlist but fails to teach or suggest the use of a computer system to test an electronically stored product configuration for errors. To the extent that this feature is not found in the prior art cited by Examiner, the present case is held allowable over the art of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Wilbert L. Starks, Jr. whose telephone number is (703) 305-0027.

Alternatively, inquiries may be directed to the following:

S. P. E. Anthony Knight	**(703) 308-3179**
After-final (FAX)	**(703) 746-7238**
Official (FAX)	**(703) 746-7239**
Non-Official/Draft (FAX)	**(703) 746-7240**

WLS

08 August 2004

Wilbert L. Starks, Jr.
Primary Examiner
Art Unit - 2121

Issue Classification	Application No.	Applicant(s)	
	09/773,101	GILPIN ET AL.	
	Examiner	Art Unit	
	Wilbert L. Starks, Jr.	2121	

ISSUE CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)	
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)
706	001	706	046
INTERNATIONAL CLASSIFICATION		703	019
G 06 N	5/02		
	/		
	/		
	/		
	/		

==========
(Assistant Examiner) (Date)

(Legal Instruments Examiner) (Date)

Wilbert L. Starks, Jr. 08 AUG 2004
(Primary Examiner) (Date)

Total Claims Allowed: 71

O.G. Print Claim(s)	O.G. Print Fig.
1	1

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	38	31	15	61		91		121		151		181
2	2	39	32		62		92		122		152		182
3	3	41	33	16	63		93		123		153		183
4	4	42	34	17	64		94		124		154		184
5	5	43	35	18	65		95		125		155		185
6	6	44	36		66		96		126		156		186
7	7	40	37		67		97		127		157		187
9	8		38		68		98		128		158		188
10	9	45	39		69		99		129		159		189
11	10	46	40	64	70		100		130		160		190
12	11	47	41	65	71		101		131		161		191
13	12	48	42	66	72		102		132		162		192
8	13	49	43	67	73		103		133		163		193
20	14	50	44	68	74		104		134		164		194
21	15	51	45	69	75		105		135		165		195
22	16	52	46	70	76		106		136		166		196
23	17	53	47	19	77		107		137		167		197
24	18	54	48	33	78		108		138		168		198
25	19	55	49	63	79		109		139		169		199
26	20	56	50	71	80		110		140		170		200
28	21	57	51		81		111		141		171		201
29	22	58	52		82		112		142		172		202
30	23	59	53		83		113		143		173		203
31	24	60	54		84		114		144		174		204
32	25	61	55		85		115		145		175		205
27	26	62	56		86		116		146		176		206
34	27	14	57		87		117		147		177		207
35	28		58		88		118		148		178		208
36	29		59		89		119		149		179		209
37	30		60		90		120		150		180		210

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (703) 746-4000

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

33438 7590 08/13/2004

HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

11/16/2004 MBEYENE2 00000103 09773101

01 FC:1501 1370.00 OP

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (703) 746-4000, on the date indicated below.

Kent B. Chambers	(Depositor's name)
	(Signature)
11-11-2004	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/773,101	01/31/2001	Kevin E. Gilpin	M-7822 US	5458

TITLE OF INVENTION: RULE BASED CONFIGURATION ENGINE FOR A DATABASE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1330	$0	$1330	11/15/2004

EXAMINER	ART UNIT	CLASS-SUBCLASS
STARKS, WILBERT L	2121	706-001000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Hamilton & Terrile, LLP
2 Kent B. Chambers
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: Trilogy Development Group, Inc.

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Austin, Texas

Please check the appropriate assignee category or categories (will not be printed on the patent); ☐ individual ☒ corporation or other private group entity ☐ government

4a. The following fee(s) are enclosed:
- ☒ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies _____

4b. Payment of Fee(s):
- ☒ A check in the amount of the fee(s) is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☒ The Director is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number 502264 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is not claiming SMALL ENTITY status. See, e.g., 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

(Authorized Signature) (Date) Nov. 11, 2004

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMIT THIS FORM WITH FEE(S)

AO 120 (Rev. 08/10)

TO:	**Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court USDC - Central District of California (Western Division) on the following

☐ Trademarks or ☑ Patents. (☐ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT USDC - Central District of California (Western Division)
PLAINTIFF Versata Software, Inc. f/k/a Trilogy Software, Inc. and Versata Development Group, Inc.		DEFENDANT Configit A/S

	PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1	6,836,766	12/28/2004	Versata Development Group, Inc.
2	10,360,612	7/23/2019	Versata Development Group, Inc.
3			
4			
5			

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director **Copy 3—Upon termination of action, mail this copy to Director**
Copy 2—Upon filing document adding patent(s), mail this copy to Director **Copy 4—Case file copy**