

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
21 December 2006 (21.12.2006)

PCT

(10) International Publication Number
WO 2006/135771 A2

- (51) International Patent Classification:
A01N 25/02 (2006.01)
- (21) International Application Number:
PCT/US2006/022528
- (22) International Filing Date: 9 June 2006 (09.06.2006)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/595,148 9 June 2005 (09.06.2005) US
- (71) Applicant (for all designated States except US):
WHIRLPOOL CORPORATION [US/US]; 2000 N M-63, Benton Harbor, MI 49022 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): MCCOY, Richard A. [US/US]; 5013 Notre Dame, Stevensville, MI 49127 (US). EBROM, Matthew P. [US/US]; 384 Wildwood Dr., Holland, MI 49423 (US). GLOTZBACH, Patrick J. [US/US]; 4447 Rachel Ln, St. Joseph, MI 49085 (US). GLOTZBACH, Mark E. [US/US]; 10446 Anderson Rd., Granger, IN 46530 (US). PAUSTIAN, Michelle A. [US/US]; 3616 Edinburgh Dr., Kalamazoo, MI 49006 (US). NIBBELINK, Matthew [US/US]; 330 Cherry St., South Haven, MI 49090 (US). MORRELL, Michael A.

[US/US]; 3256 Rollingview Ct., Hudsonville, MI 49426 (US). JENKINS, Anthony E. [US/US]; 604 W. John Beers Rd., Stevensville, MI 49127 (US). LIETO, Gregory S. [US/US]; 3719 South Ct., Muskegon, MI 49444 (US). WANG, Wei [CN/US]; 26014 Westwood Hills Dr., South Bend, IN 46628 (US). KREFMAN, Stephen D. [US/US]; 70060 42nd St., Paw Paw, MI 49079 (US).

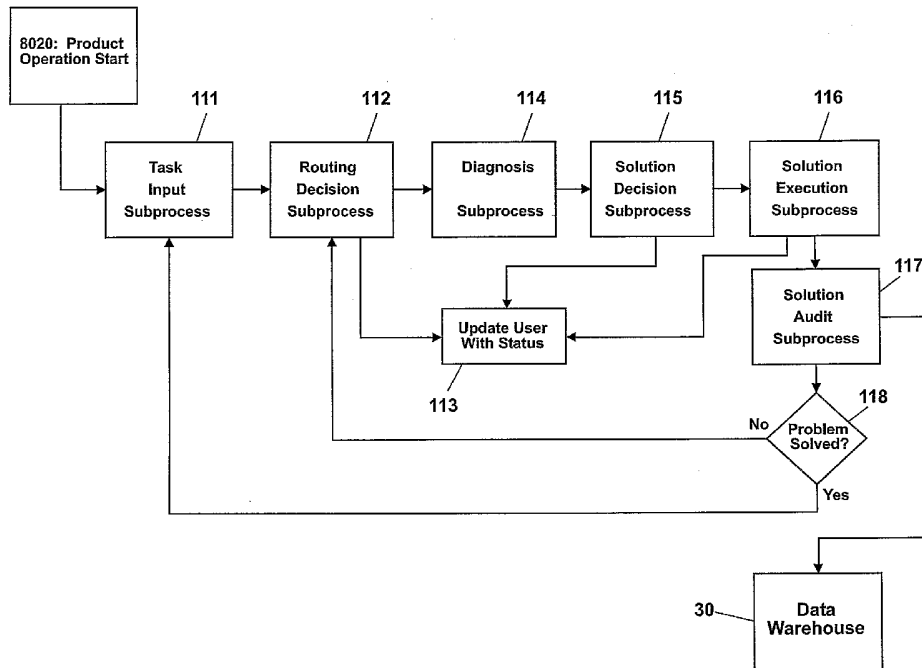
(74) Agent: DAVIS, Mark A.; McGarry Bair PC, 171 Monroe Ave. NW, Suite 600, Grand Rapids, MI 49503 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,

[Continued on next page]

(54) Title: COMPREHENSIVE SYSTEM FOR PRODUCT MANAGEMENT



(57) Abstract: A comprehensive product management system for a product is provided. The system (10) includes discrete processes for planning (1000), engineering (2000), testing (3000), producing (4000), selling (6000), using (8000), and servicing (10000) a product comprising one or more components. It also includes a comprehensive relational database (30) containing information about the product, performance, sales, consumers, and markets, and system tools (20) for operating on the discrete processes.

WO 2006/135771 A2



FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *without international search report and to be republished upon receipt of that report*

COMPREHENSIVE SYSTEM FOR PRODUCT MANAGEMENT

REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Patent Application No. 60/595,148, filed
5 June 9, 2005, the entire disclosure of which is incorporated by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates to methods for managing a product through its entire life span,
10 from its design, engineering and manufacture, through its sale, servicing, and de-
commissioning.

Description of the Related Art

Every product, whether an object, a machine, or service, and particularly a product
used in the home, has a definable life span stretching from its origin as an abstract idea to the
15 end of its usefulness. Networking technologies make it possible to systematize processes
associated with transforming a product from an idea to a useful embodiment, using,
upgrading and servicing that product during its life and decommissioning the product at the
end of its useful life.

In a world of increasing complexity and speed, there is a need to provide more
20 convenience, increased simplicity, and less cost in the delivery of products in the use,
upgrading servicing and decommissioning of those products.

SUMMARY OF THE INVENTION

A comprehensive product management system (10) includes discrete processes for
25 planning (1000), engineering (2000), testing (3000), producing (4000), selling (6000), using
(8000), and servicing (10000) a product comprising one or more components. Also provided
is a comprehensive relational database (30) containing information about the product,
including performance, sales, consumers, and markets. System tools (20) are provided for
operating on the discrete processes. Any process automatically interacts with the
30 comprehensive database, and selectively interacts with the system tools.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings:

Figs. 1 and 1A are flow charts illustrating a comprehensive system for managing a product during its life according to the invention.

5 Fig. 2 is a flow chart showing a product planning process of the comprehensive system of Fig. 1.

Fig. 3 is a flow chart showing a product engineering process of the comprehensive system of Fig. 1.

10 Fig. 4 is a flow chart showing a product and concept testing process of the comprehensive system of Fig. 1.

Fig. 5 is a flow chart showing a product manufacturing process of the comprehensive system of Fig. 1.

Fig. 6 is a flow chart showing a product marketing process of the comprehensive system of Fig. 1.

15 Fig. 7 is a flow chart showing a product sales process of the comprehensive system of Fig. 1.

Fig. 8 is a flow chart showing a product installation process of the comprehensive system of Fig. 1.

20 Fig. 9 is a flow chart showing a product registration process of the comprehensive system of Fig. 1.

Fig. 10 is a flow chart showing a product usage process of the comprehensive system of Fig. 1.

Fig. 11 is a flow chart showing a product operation process incorporated in the product usage process of Fig. 10.

25 Fig. 12 is a flow chart showing a process of making user modifications to a product in the comprehensive system of Fig. 1.

Fig. 13 is a flow chart showing a product decommissioning process in the comprehensive system of Fig. 1.

30 Fig. 14 is a flow chart showing a product service process of the comprehensive system of Fig. 1.

Fig. 15 is a flow chart showing an active diagnosis process in the product service process of Fig. 14.

Fig. 16 is a flow chart showing a PCB failure diagnosis process in the active diagnosis process of Fig. 15.

Fig. 17 is a flow chart showing a process for developing a list of components in the PCB failure diagnosis process of Fig. 16.

5 Fig. 18 is a flow chart showing a software validation process in the PCB failure diagnosis process of Fig. 16.

Fig. 19 is a flow chart showing a PCB outputs test process in the PCB failure diagnosis process of Fig. 16.

10 Fig. 20 is a flow chart showing a PCB inputs test process in the PCB failure diagnosis process of Fig. 16.

Fig. 21 is a flow chart showing a device failure test process in the active diagnosis process of Fig. 15.

Fig. 22 is a flow chart showing an appliance output device failures test process in the device failure test process of Fig. 21.

15 Fig. 23 is a flow chart showing an appliance input device failures test process in the device failure test process of Fig. 21.

Fig. 24 is a flow chart showing an automated test process in the active diagnosis process of Fig. 15.

20 Fig. 25 is a flow chart showing an ad hoc test process in the active diagnosis process of Fig. 15.

Fig. 26 is a flow chart showing an operator error test process in the active diagnosis process of Fig. 15.

Fig. 27 is a flow chart showing a network enabling process in the product service process of Fig. 14.

25

DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Figures 1 and 1A shows a comprehensive system for managing a product according to the invention. The system includes a product process 10 at one level, system tools 20 that
30 interact with the product process 10, and a data warehouse 30 that interacts with both the product process 10 and the system tools 20. It is to be understood that an exemplary product to which the product process 10 is applicable is any type of product that is usable by a

consumer, such as a tool, an appliance, a software application, or a personal service. While one or more embodiments may be described in terms of a home appliance such as a washing machine or an oven, the invention is not so limited. Moreover, it is further to be understood that while all or portions of the system and processes hereinafter described may be performed
5 automatically, e.g., by a computer or a computer network, they are not so limited. In any given application it may be cost effective for one or more portions of the system according to the invention to be performed manually.

It may be helpful to envision the system tools 20 as overlaying both the product process 10 and the data warehouse 30. The system tools 20 include a configuration editor 22,
10 a test editor 24, and a component editor 26. It will be understood that any given product will comprise one or more components, the total of which comprises a product configuration. The configuration editor 22 modifies or develops the configuration of the product. The components editor 26 modifies or develops a single component. The test editor 24 enables a testable operation of a product configuration or component. Any one of or all of the
15 configuration editor 22, the test editor 24, or the components editor 26 can engage any point of the product process 10 and/or the data warehouse 30 at any given time. While many of the functions of the system tools can be done by humans, it is within the scope of the invention for these editors to be enabled or facilitated by software. After all, capturing data for an effectively useable database such as the data warehouse 30 is accomplished best by a
20 computer network.

The data warehouse 30 is one or more relational databases containing comprehensive information about one or more products. For example the database 20000 can be a list of products to which the product process 10 is applicable. Similarly the database 21000 is
25 conceived to be a comprehensive array of product configurations, each product having one or more configurations. And database 19000 contains a catalog of components making up the product configurations of database 21000. The data warehouse 30 can also contain a list of sources 19500 for the components in the catalog. Components in appliance products, for example, will include such components and accessories as clocks, cooking aids, sensors, software, and the like as disclosed in international application entitled COMPONENTS AND
30 ACCESSORIES FOR A COMMUNICATING APPLIANCE, filed on June 9, 2006 and contemporaneously with the current application, the entire disclosure of which is incorporated

herein by reference. Components can also include such things as test protocols, specialized knowledge, user instructions, etc.

Each product in the product database 20000 will preferably be linked to a mini-warehouse 32 of data about factory tests and performance 18000, customer field tests and performance 17000, and laboratory tests and performance 16000. Each of these tests will be used to establish a database of performance benchmarks 12000. And the performance benchmark data 12000 will be related to a separate database of failure data 13000. As well, performance benchmarks 12000 will be linked to sales data 14000 which in turn is linked to consumer usage data 11000, and sales floor data 15000 (consumer feedback during the course of the sale). This linking enables a correlation between sales and performance benchmarks. In turn, consumer usage data 10000 is affected by a consumer profiles 23000 and market segmentation data 22000. Consumer profile data 23000 may be obtained, for example, from a product registration process. Market segmentation data 22000 relates to identifiable segments of known markets for the product. For example in the home appliance industry, consumers are sometimes classified into identifiable groups with similar characteristics such as "soccer moms", or "home enthusiasts."

Process adherence 30000 and data includes information on how well data from the data warehouse 30 is propagated and available to various resources of performing the product process 10. Return on investment ("ROI") 31000 data provides valuable information on the impact of reusable components on cost and the availability of resources

Effective communication between the data warehouse 30 and various aspects of the product process 10 can be accomplished by employing a software architecture that enables facile communication between internal components of a product and external components, system tools 10, resources, and the data warehouse 30. The software architecture can be implemented on and communicate over an internal communications network in the product. One example of a communications network used within an appliance product is the WIDE network protocol, created by Whirlpool, Inc., the assignee of the present patent application.

Another example of software architecture that can facilitate communication between the internal components of a product and 1) other internal components, 2) external components, 3) external components in that product or other products, or 4) other resources external to the product such as the data warehouse 30 is disclosed in International Application entitled SOFTWARE ARCHITECTURE SYSTEM AND METHOD FOR COMMUNICATION

WITH, AND MANAGEMENT OF, AT LEAST ONE COMPONENT WITHIN A HOUSEHOLD APPLIANCE, filed June 8, 2006, which claims priority on U.S. Patent Application No. 60/595,148, filed June 9, 2005, both of which are incorporated by reference. All of the methods and processes described in this application can be facilitated by the software and network structures disclosed in either of these applications.

Turning now to the product process 10 in the system according to the invention in Fig. 1, a normal starting point in the lifespan of a product is a product planning process 1000. The product planning process 1000 normally results in a product configuration 1800. It also may result in enough conceptual information to be sent to an Intellectual Property ("IP") management process 1500 where steps can be taken to commence protection of a new idea. The product configuration 1800 is then normally sent to a product engineering process 2000. Out of the product engineering process 2000, the product is normally sent to product and concept testing 3000. If the product survives product and concept testing, it is then sent to a product manufacturing process. At some point simultaneously with this path, a product marketing process 5000 is invoked. It will be apparent by the dotted lines amongst these various processes that a product need not sequentially follow the aforementioned processes, but can be fed forward at any point in the product process 10 to another point in the product process. For example, a product configuration at 1800 can be disclosed at a trade show as part of the product marketing process 5000 before it is engineered, tested, or manufactured. Similarly, prototypes out of product and concept testing can be displayed or marketed in the product marketing process 5000 to create news or preliminary demand for the product. Likewise, in any point in the path, elements of the product can be sent to the IP management process 1500.

Eventually, if the product survives to the point of being manufactured in the product manufacturing process 4000, it will enter a product sales process 6000. The product sales process 6000 will ordinarily invoke a product installation process 6500 and a product registration process 7000. These processes may be accomplished by the end user or consumer such as, for example, in the case of self-installed software or a self installed product such as a home appliance. These processes may also be accomplished by professional installers.

Normally a product will go through a product usage process 8000, during which the user may occasionally change the product by a user modification process 9000. Eventually, the product may require some sort of servicing in which case a product service of process

10000 is invoked. And, at the end of its useful life, the product in its entirety, or one or more components of the product will be de-commissioned, either voluntarily by the user or involuntarily if the product irretrievably fails.

The product planning process 1000 is more fully shown in Fig. 2. It is recognized
5 from Fig. 1 that various inputs are provided in the product planning process 1000, e.g., data from product marketing, product usage, product service, and product and concept testing. Whatever the input, the product planning process 1000 follows 1 or 2 paths, individually or simultaneously. One path contemplates the adoption of proposed new components 1010,
10 which normally go through a product and concept testing process 3000. A decision is made at 1015 whether or not to include a proposed new component in the product. If so, then the new component is added to the product configuration 1800, updating the component catalog 19000 in the data warehouse 30. It will be understood that this entire process may be conducted virtually by computer, for example.

If a proposed new component is not to be included in the product, then the process
15 returns to the start where it continues on the second path with known components 1060. Here, relevant information is retrieved from the component catalog 19000 and a decision is made at 1063 about what components to keep. The keep decision 1063 is illumined by additional data on consumer usage 11000 and sales 14000 from the data warehouse 30. A decision to keep components results in a number of planned components 1065. Typically, a
20 decision will be made at 1067 of whether or not to modify one or more planned components. The modification decision 1067 is illumined by additional data on performance benchmarks 12000, which may or may not be modified by failure data 13000. Any design modifications 1070 resulting from an affirmative modification decision are added to the component catalog 19000, and to the product configuration 1800. Typically, the new product configuration 1800
25 will be added to the product configuration database 21000, and may or may not invoke the IP management process 1500. A product successfully exiting the product planning process will normally go to the product engineering process 2000.

Fig. 3 shows the product engineering process 2000, which starts with the product configuration 1800 and whatever inputs are provided from upstream in the product process
30 10. The system does a lookup in the component catalog 19000 to determine if new components are needed at 2020 to meet the product configuration 1800. The system will also obtain data from the supply chain 32000 and sources in order to determine whether to build,

buy, or outsource a particular component. If no new components are needed then the components are configured at 2200 to the product configuration 1800. For example, components may be adapted to fit specific called or defined parameters in the product configuration 1800. At system integration 2210, the configured components are assembled
5 into a prototype, and all user interaction component 2215 is assessed. For example, in a home appliance, the process will consider how a human interacts with the product with due regard for ergonometics. What choices does the user have? Will the user interact with buttons, voice commands, dials, text screens, keyboard or the like? What options are available for wrong choices?

10 Eventually, prototypes and the product engineering process 2000 will be submitted to system tests at 2260, typically in a laboratory, logging any data for the tests at 2245, and further populating the lab test and performance database 16000. The system then compares the prototype results at 2270 to the product configuration 1800 and, decides at 2275 whether modifications are needed. If yes, the system incorporates design changes at 2290 and returns
15 to configure the components of 2200. If no changes are needed then the product is forwarded, normally to product and concept testing 3000. The IP management process 1500 will be started if not previously invoked, or updated. It will be understood that much of the product engineering process 2000 can be done either virtually by computer or physically by real-world prototypes.

20 A product and concept testing process 3000 according to the invention is shown in Fig. 4. It is contemplated that this process will apply to a specific product and ultimately provide some business model prediction with respect to that product. The process starts with a component or product configuration at 1020. Cost data can be outputted from the components or product configuration, and is variable depending upon the number of factors
25 that can be extracted from the data warehouse 30. The components and product configuration is subjected to an appropriate test retrieved from the lab test and performance database 16000. This test will output data concerning reliability, and ideally will validate the test results.

Further testing at 1025 can include employee field testing (EFT) or customer field testing (CUFT) of prototypes. Such testing will provide more data for the CUFT test and
30 performance database 17000, which will enhance information about reliability. The tests can also provide more information about consumer usage 11000 that will enable some outputs

about consumer benefits, and volume projections. As well, these tests may provide data about the manufacturability and capital requirements for the product.

Yet further testing at 1030 can include a limited manufacturing build that will provide information on factory test and performance 18000. Finally, a limited production launch in a selected market area can provide sales floor data 15000, from which the system can glean information on purchaser intent and pricing power. This, in turn, can validate information on volume projections. With a proposed component integration schedule at 1050, the system can generate a business model for the product, giving managers a valuable tool to make decisions about the product. It will be understood that product and concept testing 3000 can be done virtually by computer modeling, or in the real-world with physical prototypes and manufacturing, or a combination of both.

A product manufacturing process 4000 according to the invention is shown in Fig. 5. It typically starts with predetermined inputs as shown in Fig. 1. Although aspects of the process can easily be adapted to a service product, this process is more applicable to a physical product. For example, consider a networkable home appliance moving down assembly line. The product is assembled at 4010 to a point where some functions can be enabled. One of those functions is communication, which is enabled at 4012. A computer remote from the assembly line can attempt communication with the appliance and make adjustments necessary to assign a unique product identification at 4015. Product tests can be retrieved at 4020 from the data warehouse 30, including appropriate tests for quality control. As our home appliance moves down the line a remote test computer can identify the appliance using the pre-assigned product identification and gain access to the internal network of the appliance. The remote test computer can virtually run the appliance through its cycles according to the product configuration 1800, gain complete functionality of the appliance, and test it at 4025 according to the protocols retrieved from the data warehouse 30. The system can add the product, the product configuration, and the tested components to the respective databases 20000, 21000, and 19000, respectively, which had previously been enhanced by the product engineering process 2000, for example. Results of the tests will be sent to the factory test of performance database 18000.

A product marketing process 5000 according to the invention is shown in Fig. 6. Product marketing can start with the launch of a public relations cadence, with nothing more than concepts at 5010 to create buzz and stimulate demand, or to announce coming new

product features at 5020, or simultaneously with the launch of a product at 5015. Timing can be done relative to a product launch such as pre-launch public-relations at 5025, actual product launch public-relations at 5030, and post launch public-relations at 5040.

5 Typical public-relations types at 5050 include pilots that would be considered limited deployment of production products, partnerships such as a limited deployment with the particular partnering company, announcement of new features and accessories, special orders in special markets, and trade shows and general media. Public-relations can be centered around certain themes. For example in the home appliance industry, public relations may be centered around a green theme, i.e., how environmentally friendly the appliance is. Other
10 exemplary themes include halo ("the Angel affect"), high-tech, soccer moms, new growth (directed to financiers as an innovative company), and gadgets. The system will preferably include measurable outcomes in the product marketing process for core sales, company image, and stock price for a publicly traded company.

A product sales process 6000 according to the invention is shown in Fig. 7. The
15 product sales process 6000 starts with the consumer nearing the product at 6010. It is important to remember that the invention contemplates this nearing step occurring in the real world as a consumer approaches or passes by a product on display, and also in the virtual world as a consumer uses a mouse to click on or hover over an image in a web page. The product can be configured to solicit the consumer at 6020 directly with text, video, voice or
20 some other notification, or, alternatively, the product can be configured to notify an agent to solicit the consumer, e.g., a salesperson. At 6030, either the product itself, or the agent interacts with the consumer and gains information that can be fed to the data warehouse 30, and more particularly, consumer profiles 23000 and market segmentation 22000. As well, the interaction at 6030 can enhance the sales floor data 15000.

25 In response to that interaction or as a result of the interaction, the product or the agent at 6040 can suggest to the consumer another product. As well, the consumer can interact with the product and/or the agent to learn about the product at 6050. At some point a decision can be presented to the consumer at 6055 whether or not to purchase the product. If the consumer decides not to purchase the product, continued interaction can occur or process can terminate.
30 On the other hand, if the consumer decides to purchase the product, the consumer can be further presented at 6060 with the opportunity to customize and accessorize the product and purchase ancillary products. For example with a home networkable appliance, the consumer

can have the option to purchase extended warranties, passive diagnostic capability, and additional software enhancements. The purchase is transacted at 6070, a record of which is sent to the sales database 14000 and the process can move to installation.

A product installation process 6500 according to the invention is shown in Fig. 8. The product installation process 6500 starts with appropriate inputs (see Fig. 1.) and continues with setting the product in place at 6510. Product hookups may be required at 6520, as in the case of a home appliance. If not previously conducted in the sales process, the product can be registered at 7000. The manufacturer can make registration a condition to enable usage of the product at 6530, or, alternatively permit the consumer to bypass registration and provide periodic reminders at 6535. It is contemplated that enablement, particularly of an appliance, can include various modes, any one or all of which can be enabled depending on what was purchased. For example consider a networkable washing machine appliance. Installation can include enabling an automated use and care guide at 6540, which, on activation during usage, can automatically provide instruction to the consumer on use of the appliance. When the consumer pushes a particular button on the washing machine, the appliance can issue a voice suggestion or provide text display that only certain detergents will work with the particular feature selected. Installation can also include enabling passive diagnostics at 6550. Here, the appliance can continuously monitor the performance of all components of the washing machine and advise the consumer (again by voice or by text) that a particular component is about to fail and needs to be replaced. Installation can also include enabling information mediation at 6560 where the consumer is presented with the opportunity to share usage data of the washing machine with the manufacturer or with a third-party. Installation can also include enabling an enhancement guide where the consumer can be presented with reminders of available enhancements, based on automatically observed interactions between the consumer and the washing machine. Installation can also include enabling enhancements to the networkability of the washing machine. For example, the unique identifier for the washing machine can automatically be added to the list of recognized devices in other networked appliances. Moreover, the washing machine can be configured to automatically acquire identifiers in later acquired appliances and coordinate automatically updated ownership information with the data warehouse 30 for previously registered owners.

A product registration process 7000 according to the invention is shown in Fig. 9 and starts by invoking the process at 7010. As previously indicated, this can occur in the product

sales process 6000, in the product installation process 6500, separately or sometime during the product usage process 8000. As well, product registration can occur at the user modification process 9000, or even during product service 10000 (see Fig. 1). Product registration can be invoked at the consumer command, or the command of the sales agent, or automatically with enablement of the product. Once invoked, a product identification is uploaded at 7020 and any appropriate customer identification is downloaded. That information can be gleaned from other products or home devices at 7025 or from sales data 14000. In this embodiment, the product has a GPS installed to automatically record an address where the product is installed. Continuous transmission of the address to the data warehouse 30 can enable theft control if the product is portable and subject to theft. For example, a portable microwave moved from its registered address can trigger a notification to the consumer (e.g. by cell phone or email) that the appliance has been moved. Moreover, continuous transmission can track the location of the microwave during its movement.

The consumer is preferably given the opportunity to confirm and correct retrieved data at 7030. Once confirmed, the product can be enabled for usage at 7060 (if registration is a condition for enablement), and the warranty can be activated at 7050. Preferably, warranty and service databases at the data warehouse 30 will be updated at 7040. In the case of a product that may be part of an energy management arrangement (discussed below), notification can automatically be sent to an appropriate energy utility at 7200 and this consumer can be enrolled in a resource management program 7210.

A product usage process 8000 according to the invention is shown in Fig. 10. Product usage starts with the product operation 8020 according to the invention and can present the user with some of the options available during product installation 6500. For example, the consumer can be presented with a decision at 8015 to share usage data of the product with the manufacturer or with a third-party. If the decision is affirmative, the product is configured into an information mediation system at 8010 and that information is transmitted to the appropriate database at the data warehouse 30. Similarly, the consumer can be presented with the decision at 8025 and enable an automated use and care guide which, on activation during usage at 8030, can automatically provide instruction to the consumer on use of the product. As well, at any point during product operation, enhancements can be suggested to the consumer or by the consumer at 8030. If a particular feature is needed, and enhancements process 9000 is invoked.

The product operation process 8020 according to the invention is shown in Fig. 11. Here, product operation commences with the task input 111. The task input 111 can be a sub process unto itself, and may include something as simple as a button push, a switch operation, a text input, or voice command. It can automatically be invoked by a software command or the automatic consequence of a diagnosed problem. Normally the task input 111 will result in some routing decision 112 where the product will be called upon to resolve the task input directly, through some external agent or client, or through human intervention. Preferably, some progress indication 113 can be provided to the consumer from the routing decision 112. For example, in the cooking appliance such as an oven, the status of the oven can be updated such as a countdown timer starting before cooking actually begins.

If appropriate enhancements have been enabled on the product, a diagnosis sub process 114 can be invoked to provide some contextual response to the input command and routing decision, or refine the command and routing based on the context. For example, a washing machine can be programmed to accept commands only from an authorized user. The diagnosis sub process 114 can interrogate the entry of the task input at 111 to ensure that the command came from an authorized user. The diagnosis sub process can also identify a problem with the task input and provide an alternative solution decision 115. For example, say the consumer entered a command into the washing machine for a small wash, but overloaded the washing machine with clothes. The washing machine can detect the overload, determined an appropriate solution and suggests a solution to the consumer (remove some clothes), or even automatically override the task input for a small wash with a command for a large wash, notifying and updating the user with the status at 113. The solution is redefined and executed at 116 and audited at 117. Preferably the data is logged and up loaded to the data warehouse 30, and the user is updated with the actual result of the task input. If the result diverges from the original task input, the product can provide some context or reason for the divergence, and suggest tips for avoiding divergence in the future.

If a problem is detected, the user is preferably presented with an inquiry at 118 about whether or not the problem was solved. If not, the routing decision 112 is reevaluated, and if so, the system is reset for entry of a new task input at 111. It is to be remembered that the diagnosis sub process 114 does not only refer to diagnosing problems, but more broadly the diagnosing of alternative commands. For example, the process could result in a recommendation to the user of a different command, automatic implementation of a different

command, an error message, with or without implementation, a helpful hint or warning, and offer to purchase a new product or enhancement, or an offer to participate in the aforementioned information mediation system or energy management system. The diagnosis can be based on user preferences or experience from previous or continued audits, energy
5 consumption, resource availability, timing based on noise and vibration, timing based on use or demand, sensors and other inputs, availability of cycles, and information exchanged on the network with other products.

The process of Fig. 11 can be used as a template to implement any of the processes disclosed herein. That is, whatever the process or process step might be, it can be resolved to
10 or thought of as a task input. The task input is then routed for decision, especially whether the task can be resolved locally, by human intervention, or by an external agent. In the case of the product, the product will look for a local solution within the product, to the user, or to a source exteriorly of the product. In the diagnosis step, the task will be analyzed or interrogated to ensure the validity of the command. If need be the task is modified,
15 alternatives can be suggested, or a new solution to the task can be proposed. The resulting solution to the task is then executed and subject to an audit. As one can readily see, when the process of Fig. 11 is applied to a process, it provides for: the inspection of the process and all of its steps; a determination of where the steps are to be completed or where information to complete the steps is to be drawn, confirmation of the appropriateness of the steps,
20 modifications to the steps, or alternatives to the steps; the implementation of the steps; and the auditing of the solution. The result and related information is then saved to the Data Warehouse 30 for subsequent use. The process of Fig. 11 is repeated until the problem is solved. In this way the process can self-improve. The template can be applied to the process as whole, a subset of the steps, on a step-by-step basis, or a combination of them all.

25 A process for user modifications 9000 according to the invention is shown in Fig. 12. This process contemplates enabling the user to add or remove components in the product. Modifications to be added might have been purchased by the consumer in the sales process 6000 or initiated in the product usage process 8000. The process commences with the installation of the modification at 9010. It will be understood that in this context, installation
30 may also include removal of a component. The product or a system of communications in which the product is networked may have to suspend normal operation at 9010 during installation, and the modification joins the system where the system recognizes the

modification. Simultaneously, product registration in accord with the product registration process 7000 can occur. It may be necessary to validate the modification at 9025 prior to configuration. Validation can include confirmation that the modification is appropriate, authorized, registered, and operable. If not validated, a notification at 9027 can update the consumer usage data base 11000 at the data warehouse 30. If validated, the modification either can configure itself or allow the system to configure itself at 9030, or the system adapts to the modification and allows the modification to configure the system at 9040, or some combination of both. Preferably, the data warehouse 30 is updated with consumer usage 11000, service and warranty data 33000, and product registration data 34000. Once configured, the product resumes normal operation at 9050 and the system interacts with the modification at 9060.

An example of a product enhancement is the registration of the product with a resource provider, such as an electrical utility. The user can register online with the resource provider and contract for purchasing electricity at predetermined rates, which can include reduced rates for non-peak usage. The user could also contract for electrical supply to be terminated or reduced by the resource provider based on system demands. The user could also contract for the supply of energy based on a commodities market price approach, the price of which could be set by online auctions.

A product decommissioning process 9500 according to the invention is shown in Fig. 13. At some point the useful life of the product or component of the product will come to an end. This begins the product decommissioning process 9500. The consumer may decide to remove the product from use at 9600. Preferably, the consumer need only inform one of the network products of the decommissioning of an identified product at 9610. The consumer can be prompted to power down the product or remove communication and 9620. In another aspect of this process, products on the network may sense at 9510 that a product is missing from its registry of connected devices, also known as a "know about list." The networked system makes a determination at 9520 of the product identified to be missing. An inquiry at 9525 to the consumer verifies whether the removal is temporary or permanent. If permanent, the networked system removes the product from the know about list at 9530, prompts the user with recycle information and 9540 or suggested assistance for de-installation of the product, and may suggest a replacement or provide for automated ordering and 9550. Preferably, the warranty and service databases at the data warehouse 30 are updated at 7040.

A product service process 10000 according to the invention is shown in Fig. 14. The product or service process 10000 provides one of the most convenient benefits of the comprehensive system of product management according to the invention. The product service process commences with an inquiry at 12 of whether or not passive diagnosis in the product is enabled. If not enabled, the consumer is presented with another inquiry at 13 with the opportunity to install a passive diagnosis component. If the answer to that inquiry is yes, passive diagnosis component is installed and commenced. If the answer that inquiry is no, the user is prompted with another inquiry at 22 about whether a problem is observed. If no problem is observed, the system restores to the product service start. If a problem is observed, then the user is presented with another inquiry at 24 about whether the product is network ready. If not, the user is prompted at 25 to enable the network using the network enabling method at 26. If the user chooses not to enable the network, the user can be presented with a use and care guide, warranty information, or other service information at 27.

Once the network is enabled, the system invokes one of three active diagnosis processes, as preferred by the consumer. The active diagnosis processes include do-it-yourself (DIY) service at 30, remote service at 40, or local service at 50. The consumer can be presented with a preference list, menu, or sequential inquiries. These choices are directed to who is going to conduct the active diagnosis process at 60. It is contemplated that a product capable of being networked will have one or more printed circuit boards (PCB's) and input/output (IO) ports. Service architectures 32 will be predetermined for the product and will comprise ways of communicating problems in getting solutions during the diagnosis process. Exemplary service architectures include (1) integrated audible codes on a PCB in the product that can be communicated to a service center by telephone (wireless or land line), (2) integrated audible codes on multiple PCBs in the product networked together that can be communicated to a service center by telephone (wireless or land line), (3) distributed audible codes on multiple PCBs in the product networked together that that can be communicated to a service center by telephone (wireless or land line), (4) external audible signals wirelessly connected to one or more PCBs that can be communicated to a service center by telephone (wireless or land line), (5) a service computer connected by cable to one or more PCBs in the product by way of an IO port, (6) a service computer connected wirelessly to one or more PCBs in the product by way of an IO port, (7) a remote data connection by Internet or phone modem between a service center and an IO port on the product, (8) a remote data connection

by USB, flash drives and automated keys, conveyed by courier, (9) error codes displayed on board the product, (10) error codes displayed on board a remote product networked with the diagnosed product, and (11) the traditional backend technician communicating with consumer or service tech.

5 The active diagnosis process 60 is operated as described below, and the consumer is then presented with an inquiry at 62 of whether or not the solution was identified. If not, the active diagnosis process is recommenced. If so, the system can identify parts or replacements needed at 70, automatically order parts at 72, order replacement and 74, received the parts or replacement at 73, and execute a solution at 74.

10 If the passive diagnosis process operates at 14, it faces an inquiry at 20 of whether not problem is detected. If the problem is not detected the consumer is presented with the inquiry 22 about observation of the problem. On the other hand if the passive diagnosis process detects a problem, the system notifies the consumer at 15 and faces an inquiry at 28 of whether or not a solution as identified. If the solution is automatically identified, the system's
15 proceeds to ordering parts or replacements at 70. If the solution is not identified a 28, the active diagnosis process is invoked.

 Upon execution of the solution at 74 the system then queries at 75 whether or not a warranty is in place, gathering pertinent termination from the data warehouse 30. If a
warranty is in place, the warrantor bears responsibility for the cost of the solution. If no
20 warranty is in place, the user bears responsibility for the cost of the solution.

 The active diagnosis process 60 is shown in Fig. 15 with the sequential tests in the process shown in Figs. 16 through 26. The active diagnosis process 60 commences with establishing communication at 601 between a test observer and the product. Also, at 603, the data warehouse 30 is accessed. The test observer at 602 acquires the product's or
25 component's identification and, at 604, based on that identification retrieves appropriate software, data, documents, applications, and components and product information from the data warehouse 30. Depending on the scope of a failure, for example as in a total failure mode, the test observer may be prompted to change the service architecture 32 for the diagnosis.

30 The first test in the active diagnosis process is the PCB failures test at 630. The PCB failures diagnosis is shown in Fig. 16, and commences with the test observer developing a list of components to operate on. Looking now to Fig. 17, the list of components 632 is obtained

by executing a discovery command at 634 to retrieve appropriate data on the components from the data warehouse 30 or from stored data in the product. Appropriate component identification is stored at 636 in an appropriate memory location. Returning to Fig. 16, the PCB failure diagnosis proceeds to an inquiry at 635 to determine if all components in the list
5 have been tested. If not, then the system proceeds to a software validation at 640.

The software validation process 640 is shown in Fig. 18. The process commences at 641 with a check for any updated software versions available for the PCB component or the product. The system compares product and components version information at 604 with updated product configuration data 21000 from the data warehouse 30, and answers an
10 inquiry at 642 whether an update is required. If an update is required, the update is performed at 645. If an update is not required, the system is presented with an inquiry at 643 about service bulletins. If service bulletins are available and needed, appropriate action is taken at 646 to obtain them. If no further service bulletins are available or needed, the system then conducts a cyclical redundancy check (CRC) at 644 and compares the expected CRC for the
15 product to the actual CRC. An inquiry at 647 determines if there is a mismatch in the compares. If there is a mismatch, then the PCB component is determined to have failed and the PCB failure diagnosis process returns a failure. If there is no mismatch, then the PCB failure diagnosis process returns a pass and proceeds to a PCB outputs test 650 (see Fig. 16).

The PCB outputs test process 650 is shown in Figs. 19-19B. The process commences
20 with a reconfiguration at 652 of the PCB connections according to Figs. 19A and 19B. The wiring harness 104 between the PCB 100 and the devices 102 that are controlled by the PCB as shown in Fig. 19A is disconnected and reconnected to a test fixture 101, operably connected to the test observer at 105 as shown in Fig. 19B. Typically, this reconnection will be done manually, but it is within the scope of the invention for the reconnection and
25 reconfiguration to be done electronically by way of a switch, or electronically for example. The connection between the test fixture 101 and the test observer 105 can be wireless or remote, depending upon the service architecture 32 applied. The test fixture 101 can be little more than an IO port on the product. With access to the product configuration data 21000 and of the product or complement version information 604, the test observer is configured at
30 653. At 654 the test observer energizes each output, individually and sequentially, sensing the corresponding power output at 656 and comparing the result to the expected results based on information in the product configuration data 21000. Each test results in an inquiry at 658

of whether or not a mismatch has been detected. If there is a mismatch, then the PCB component is determined to have failed and the PCB failure diagnosis process returns a failure. If a mismatch is not detected, then the system inquires whether or not all outputs have been tested at 659. If all outputs have not been tested, the system loops to the next
5 output. If there is no mismatch on any output, then the PCB failure diagnosis process returns a pass and proceeds to a PCB inputs test 660 (see Fig. 16).

The PCB inputs test process 660 is shown in Figs. 20-20B. The process commences with a reconfiguration at 662 of the PCB connections according to Figs. 20A and 20B. The wiring harness 104 between the PCB 100 and the devices 102 that are controlled by the PCB
10 as shown in Fig. 20A is disconnected and reconnected to a test fixture 101, operably connected to the test observer at 105 as shown in Fig. 20B. Typically, this reconnection will be done manually, but it is within the scope of the invention for the reconnection and reconfiguration to be done electronically by way of a switch, or electronically for example. The connection between the test fixture 101 and the test observer 105 can be wireless or
15 remote, depending upon the service architecture 32 applied. The test fixture 101 can be little more than an IO port on the product. With access to the product configuration data 21000 and of the product or component version information 604, the test observer is configured at 663. In addition, at 663A, the test observer is also configured to access specific memory locations on the PCB. The data acquisition engine disclosed in either international
20 application entitled SOFTWARE ARCHITECTURE SYSTEM AND METHOD FOR COMMUNICATION WITH, AND MANAGEMENT OF, AT LEAST ONE COMPONENT WITHIN A HOUSEHOLD APPLIANCE, filed June 8, 2006, which claims priority on U.S. Patent Application No. 60/595,148, filed June 9, 2005, is particularly adapted to enable such a configuration in the test observer. At 664, the test observer sets each input according to the
25 product configuration 21000 as if the product were operating, attempting to drive the PCB to an expected result. Consequently, at 666 the test observer measures a value at a corresponding memory location and compares the measured value to the expected value. Each test results an inquiry at 668 of whether or not a mismatch has been detected. If there is a mismatch, then the PCB component is determined to have failed and the PCB failure
30 diagnosis process returns a failure. If a mismatch is not detected, then the system inquires whether or not all inputs have been tested at 669. If all inputs have not been tested, the

system loops to the next input. If there is no mismatch on any input, then the PCB failure diagnosis process returns a pass and proceeds to a device failure test 670 (see Fig. 15).

The second test in the active diagnosis process is the device failures test process 670. The device failures diagnosis is shown in Fig. 21, and commences with the test observer
5 developing a list of components to operate on. See Fig. 17. The device failures diagnosis proceeds to an inquiry at 671 to determine if all components in the list have been tested. If not, then the system proceeds to a product output device failures test at 670A.

The product output device failures diagnosis is shown in Fig. 22. It commences at
10 672 with an inquiry about whether or not power sensing is enabled. In order for this diagnosis to be effective, the system must provide some way of sensing power such as by way of current, either individually or at a common line. Appropriate information on power sensing may be obtained from the product configuration data 21000 or data warehouse 30. If power sensing is not enabled, the power sensing apparatus must be configured at 678. If power sensing is enabled, that the second inquiry is presented at 673 to determine whether or
15 not component self test is enabled. This inquiry answers the question of whether or not the expected power curve is present for comparison or whether it needs to be obtained. If the component self test is not enabled, the test observer must be configured at 678 with the appropriate device and device power consumption profiles at 673, obtained from the product configuration data 21000. Once everything is configured, the product output device failures
20 diagnosis proceeds to an inquiry at 674 to determine if all devices in the list have been tested. If not then the next device output is set to on at 675 and the actual power consumption profile is compared at 676 to the expected power consumption profile. If there is a mismatch, then the PCB component is determined to have failed and the PCB failure diagnosis process returns a failure. (See Fig. 15) If none of the devices show a mismatch, and the process
25 returns to the product input device failures test at 670B (see Fig. 21).

The product input device failures diagnosis is shown in Fig. 23. It commences at 700 with an inquiry about whether or not component self test is enabled. If not, then the test observer and the PCB are configured at 702 with appropriate plant models obtained at 704
30 from the product configuration database 21000. Plant models drive the input to an expected result or state at the output. The system determines and configures appropriate data collection at 708 and sets an output or memory pattern according to the input device plant model definition obtained from the plant model at 712. The process then determines at 713 if virtual

key presses or some other virtual input is required in order to drive the input to the expected result. If so then virtual commands are set at 714 and if not, in inquiry presented at 715 of whether user intervention is required. An example of user intervention in a washing machine might be to manually insert an unbalanced load. If user intervention is required, then the user is prompted at 716 to initiate an action in the system inquiries at 717 whether the action has been completed. If so, then the input value is compared to the expected output result at 718. If there is a mismatch, then the PCB component is determined to have failed and the PCB failure diagnosis process returns a failure. (See Fig. 15) If the device does not show a mismatch, then the process returns to an inquiry at 710 of whether or not all devices have been tested. If not, then the next input device is set at 712 and the test redone. If there is no mismatch on any input device, then the device failure diagnosis process returns a pass and proceeds to an automated test 610 (see Fig. 15).

The automated test process 610 is the third test and is shown in Fig. 24. This diagnosis checks for more subtle issues such as environmental issues, overheating, or types of consumables, and draws more extensively on statistical evidence at the data warehouse 30. The automated test diagnosis is shown in Fig. 24, and commences at 611 with an inquiry about whether or not the component self test is enabled. If not, then the test observer is configured at 612 with system tests macros 613 obtained from the product configuration database 21000. The system determines and configures appropriate data collection at 614 and retrieves the macros' expected data patterns at 660. An inquiry at 618 asks whether or not all macros have been completed. If not, then the process sets outputs and memory values according to the next macro definition at 620 and checks the input value against expected results. If there is a mismatch, then the PCB component is determined to have failed and the PCB active diagnosis process returns a failure. (See Fig. 15) If the device does not show a mismatch, the process returns to the inquiry at 618 of whether not all macros have been tested. If there is no mismatch on any macro, then the automated test diagnosis process returns a pass and proceeds to another test in the active diagnosis process (see Fig. 15).

The active diagnosis process contemplates the PCB failures diagnosis 630 and the device failure diagnosis of 670 in the automated test diagnosis 610 as being conducted sequentially. Two more tests are available in the active diagnosis process, and ad hoc test 690 and an operator error test 720, including initially before the first three, or after.

The ad hoc test 690 is shown in Fig. 25. This test is designed to empower the user with enough information to capture something new in a diagnosis of the product. It commences with the configuration of the test observer at 692 based on data from the product configuration database 21000. It commences downloading engineering documents at 694, service history of the product, including model and class and 696 and in downloading previous ad hoc transaction logs if any are applicable. At 693, the test observer enters a process capture mode and prompts the user at 695 to capture certain user information. Once all the information is in place, the system queries at 698 for additional ideas for diagnostic tests. At this point the user can select one or more test ideas from the download or add new tests to execute. An affirmative response to the inquiry about additional tests to be conducted results in another inquiry at 699 about whether or not the test already exists. If so, then the system proceeds to execute the tests at 708. If the test is new, the new test is created at 700, test steps are queried and added at 702, 704, and the system automatically assigns a globally unique identifier to the test at 706. One or more tests are executed at 708, with captured data resultant and perhaps the test observer's analysis of the results. The test is logged 710 and submitted to the data warehouse 30. An inquiry 712 determines whether or not the problem has been identified and whether or not the ad hoc test is continued or returned to the active diagnosis process 60.

The operator error test 720 is shown in Fig. 26. This test commences with several inquiries designed to determine the level of engagement with the user. For example an inquiry at 722 determines whether of the product is speech enabled, and at 724 whether or not the product is text enabled, and at 725 whether or not the product is video enabled, and at 726 whether or not the user desires to purchase a kit to enhance or enable such communication. If so the modifications downloaded at 728, with appropriate data from the product configuration database 21000. Once communication is established, the problem is inputted at 730 and an inquiry at 732 determines if the problem is recognized. If so, the system compares user settings and actions in the product to a preferred model inquiries a mismatch at 736. If a mismatch exists, the suggested alternate selection is prompted to the user at 734 whereupon selections can be changed at 740 and an inquiry about resolution of the problem at 742. If the problem is resolved the transaction is logged at 740 and that diagnosis is terminated. If the problem is not resolved, a further problem can be inputted or the problem can be re-inputted at 730. If there is no mismatch at 736, the system can prompt some interaction with the user

at 738. Failure of further interaction at 744 will move the system toward some help mode at 750. The user can determine at 752 if further interaction with the user interface is desired. If not the transaction ends and is logged 740. If so, output information is provided at 754 and an interaction with the user is conducted at 756 and the problem may or may not be resolved at 758. If the problem is not solved, the process loops back to step 752. If the problem is resolved, the transaction is logged and control returns to step 720 to resolve any further errors, and, if none, terminated.

Referring again to Fig. 14 and the product service process 10000, the network enabling method 26 is shown in Fig. 27. It commences with an inquiry at 262 about whether a network connection is available. If not, then user is prompted at 264 to purchase or otherwise acquire and install a connection kit. If a network connection is available, or if the user has otherwise installed a network connection, the system queries at 266 whether or not the network connection is wired or wireless. If wireless, the user is presented with the opportunity at 268 to acquire, such as by purchase, a key, a dongle, or other accessory to enable and facilitate the active diagnosis process. Software is installed, commissioned and operated with the goal of establishing network communication with minimal crosstalk among possibly conflicting devices.

Referring again to Fig. 14 and the passive diagnosis process at 14, it will be understood that the passive diagnosis process will be similar to or the same as the active diagnosis process but configured to run passively without user or test observer intervention. In such an embodiment, the test observer will be built in to the product, remote or automatic switching of connections can occur by software command, and the time to run such an automatic passive diagnosis will normally be at the time when the product is not in operation.

While the invention has been specifically described in connection with certain specific embodiments thereof, it is to be understood that this is by way of illustration and not of limitation, and the scope of the appended claims should be construed as broadly as the prior art will permit.

CLAIMS

What is claimed is:

1. A comprehensive product management system for a product comprising one or more components, the system comprising discrete processes for planning, engineering, testing, producing, selling, using, and servicing a product, a comprehensive relational database containing information about the product, performance, sales, consumers, and markets, and system tools for operating on the discrete processes, wherein any process automatically interacts with the comprehensive database, and selectively interacts with the system tools.
2. The system according to claim 1 and further comprising at least one of the following: product planning cycle, IP management cycle, product engineering cycle, product testing cycle, concept testing cycle, product manufacturing cycle, product marketing cycle, product sales cycle, product installation cycle, product registration cycle, product usage cycle, product operation cycle, product enhancements cycle, product decommissioning cycle, product service cycle, and product diagnostic cycle.
3. The system according to claim 2, wherein the product diagnostic cycle comprises at least one of: a PCB failure cycle, sequential component test cycle, software validation cycle, PCB output test cycle, PCB input test cycle, input device test cycle, output device test cycle, automated system test cycle, ad hoc test cycle, operator error cycle, and user instruct cycle.
4. The system according to claim 2, wherein the product test cycle comprises at least one of: physical prototype test cycle, virtual prototype test cycle, and customer field test cycle.
5. The system according to claim 2, wherein the product manufacturing cycle comprises at least one of: product identification cycle, product communication cycle, and product virtual test cycle.
6. The system according to claim 2, wherein the product marketing cycle comprises at least one of: product idea cycle, product feature cycle, product pre-launch cycle, product launch cycle, and product post-launch cycle.

7. The system according to claim 2, wherein the product sales cycle comprises at least one of: consumer solicitation cycle, consumer interaction cycle, consumer feedback cycle, warranty cycle, and product customization cycle.

8. The system according to claim 2, wherein the product installation cycle comprises at least one of: resource hookup cycle, product registration cycle, and enable product usage cycle.

9. The system according to claim 2, wherein the product registration cycle comprises at least one of: product identification cycle, customer identification cycle, product location cycle, user feedback cycle, warranty activation cycle, and resource registration cycle.

10. The system according to claim 2, wherein the product operation cycle comprises at least one of: task input cycle, task decision cycle, task diagnosis cycle, task solution cycle, solution execution cycle, and solution audit cycle.

11. The system according to claim 2, wherein the product enhancements cycle comprises at least one of: modification installation cycle, operation suspension cycle, and product configuration cycle.

12. The system according to claim 2, wherein the product de-commissioning cycle comprises at least one of: identification of de-commissioned product cycle, and product replacement cycle.

13. The system according to claim 2, where in the product service cycle comprises at least one of: do it yourself service cycle, remote service cycle, local service cycle, active diagnosis cycle, and passive diagnosis cycle.

14. A comprehensive product management system for a product comprising one or more components, the system comprising:

discrete processes for planning, engineering, testing, producing, selling, using, and servicing a product,

5 a comprehensive relational database containing information about the product, performance, sales, consumers, and markets, and

system tools for operating on the discrete processes, wherein any process automatically interacts with the comprehensive database, and selectively interacts with the system tools.

15. The system according to claim 14 wherein at least a first plurality of the discrete processes provide experiential information to the database relating to plurality of products and at least a second plurality of discrete processes for a first product accesses information from the database provided by a different one of the discrete process relating to a
5 second product as a basis for making a control decision for the first product.

16. The system according to claim 14 wherein the system tools comprise a configuration editor adapted to configure the product, a component editor adapted to configure at least one component of the product. and a test editor adapted to enable a testable operation of at least one of the product and the component.

17. The system according to claim 14 wherein the database comprises at least one relational database containing at least one of:
a catalogue of components potentially applicable to the product,
performance benchmark data,
5 consumer data, and
ROI data.

18. The system according to claim 14 wherein the performance benchmark data comprises at least one of:
factory test and performance data,
consumer field test and performance data, and
5 laboratory test and performance data.

19. The system according to claim 14 wherein the customer data comprises at least one of:
customer usage data,
sales floor data, and
5 customer profile data.

20. The system according to claim 14 further comprising at least one of a discrete processes for marketing, installation, product registration, product usage, user modification and decommissioning.

21. The system according to claim 14 wherein at least one of the discrete process comprises the steps of:

locally inputting a task,

deciding where to route the decision making from a selection of at least local, user and

5 remote

diagnosing the task,

selecting a decision,

executing the decision, and

auditing the decision.

22. A comprehensive product management method for a product comprising one or more components, the method comprising the steps of

planning the product while automatically interacting with a comprehensive relational database containing information about the product, performance, sales, consumers, and

5 markets;

engineering the product while automatically interacting with the database;

testing the product while automatically interacting with the database;

producing the product while automatically interacting with the database;

selling the product while automatically interacting with the database;

10 operating the product while automatically interacting with the database; and

servicing the product while automatically interacting with the database.

23. The method according to claim 22 further comprising at least one of :

marketing the product, while automatically interacting with the database;

installing the product, while automatically interacting with the database;

registering the product, while automatically interacting with the database,

5 operating the product, while automatically interacting with the database;

modifying the product, while automatically interacting with the database; and

decommissioning the product. while automatically interacting with the database.

24. The method of claim 23 wherein the operating of the product comprises at least one of:

locally inputting a task,

deciding where to route the decision making from a selection of at least local decision

5 making, user decision making and remote decision making,

diagnosing the task,

selecting a decision,

executing the decision, and

auditing the decision,

10 wherein remote decision making uses a comprehensive relational database containing information about the product, performance, sales, consumers, and markets, and system tools for operating on the discrete processes.

25. A comprehensive product management method for a product comprising one or more components, the method comprising the steps of

locally inputting a task selected from a testing task, an operating task, a servicing task servicing the product, marketing task and installation task, a registering task, a modifying task
5 and a decommissioning task,

deciding where to route the decision making from a selection of at least local decision making, user decision making and remote decision making,

diagnosing the task,

selecting a decision,

10 executing the decision, and

auditing the decision,

wherein remote decision making uses the database and system tools to implement the diagnosing, selecting and auditing steps.

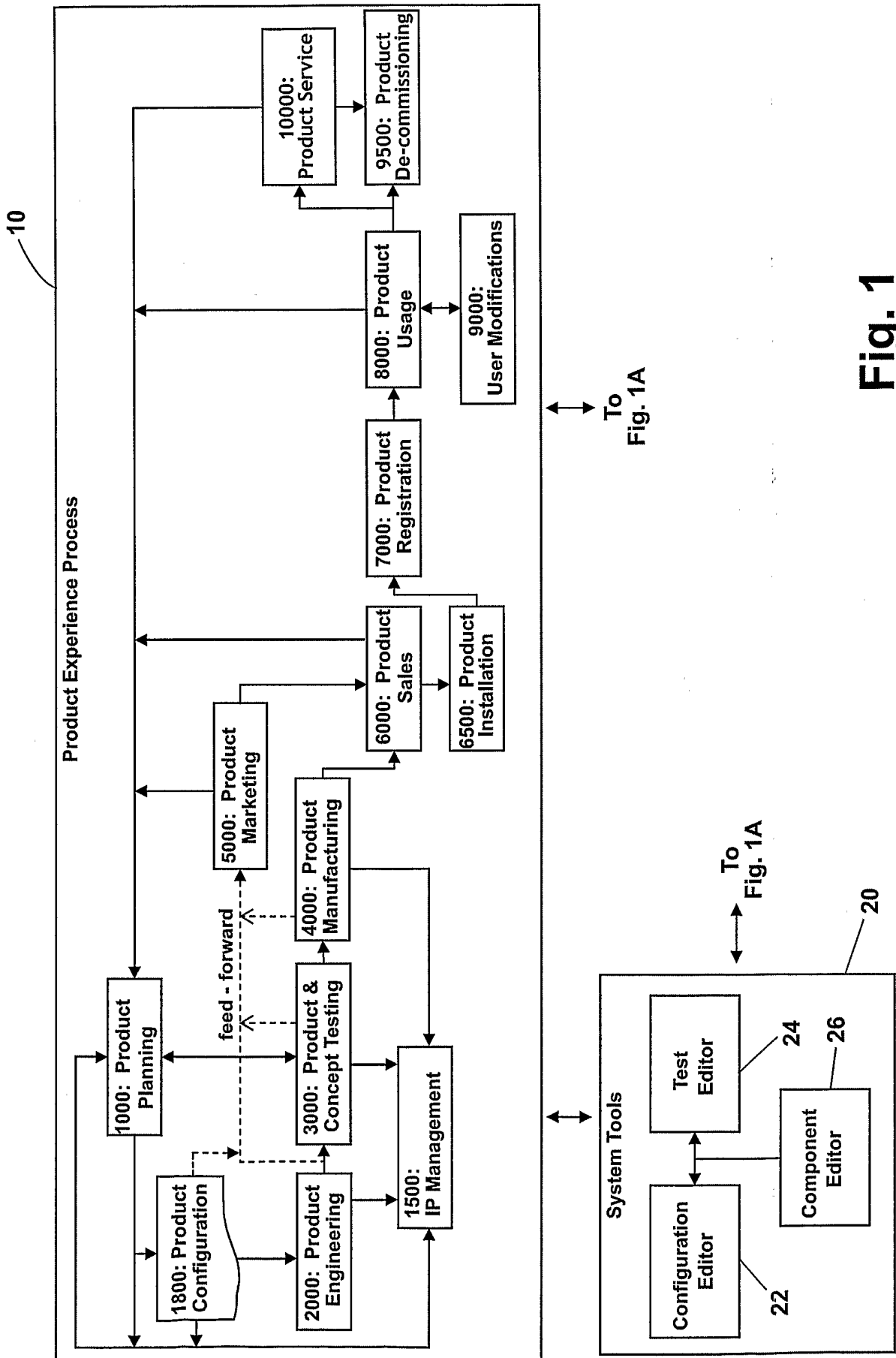


Fig. 1

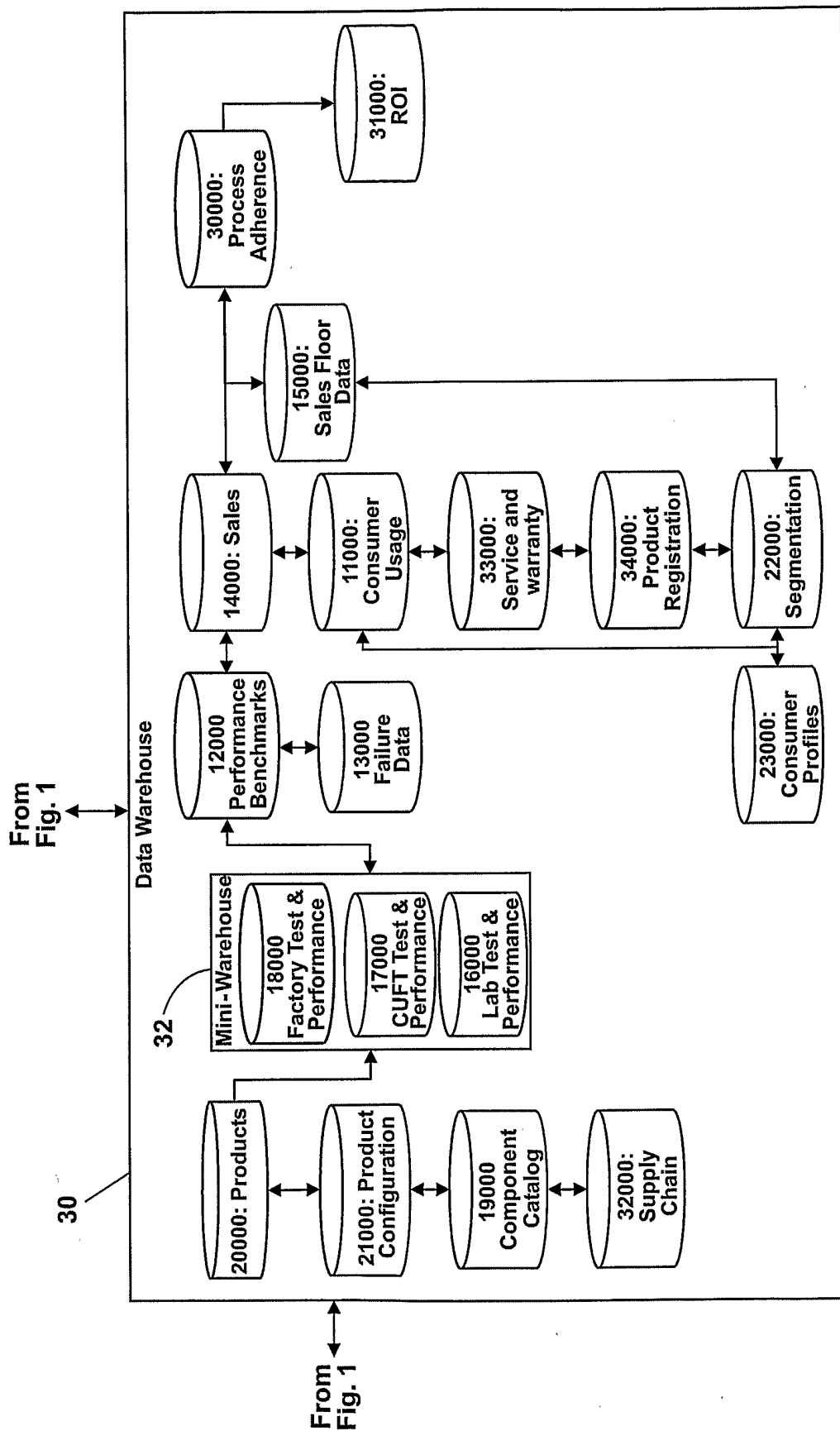


Fig. 1A

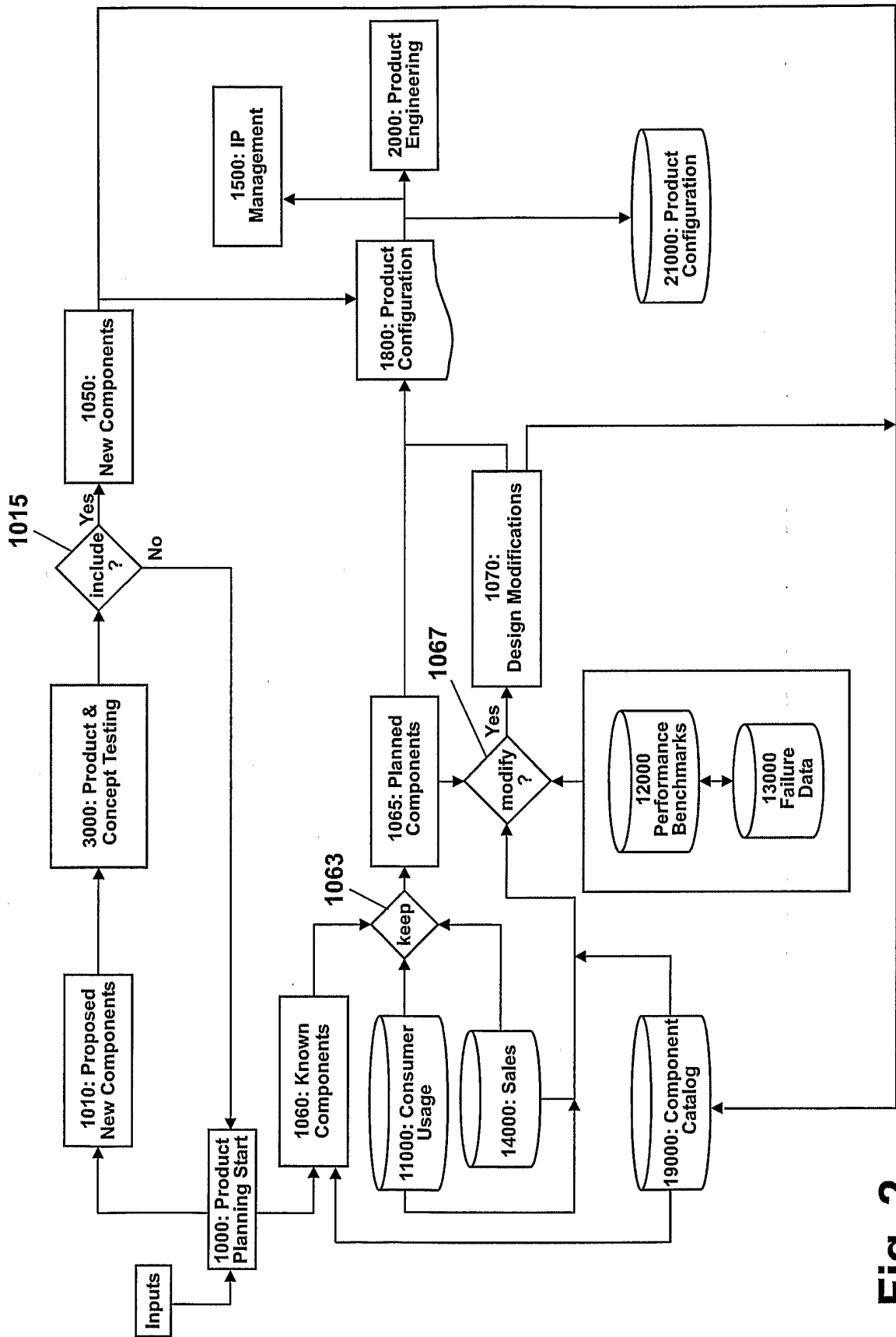


Fig. 2

4/28

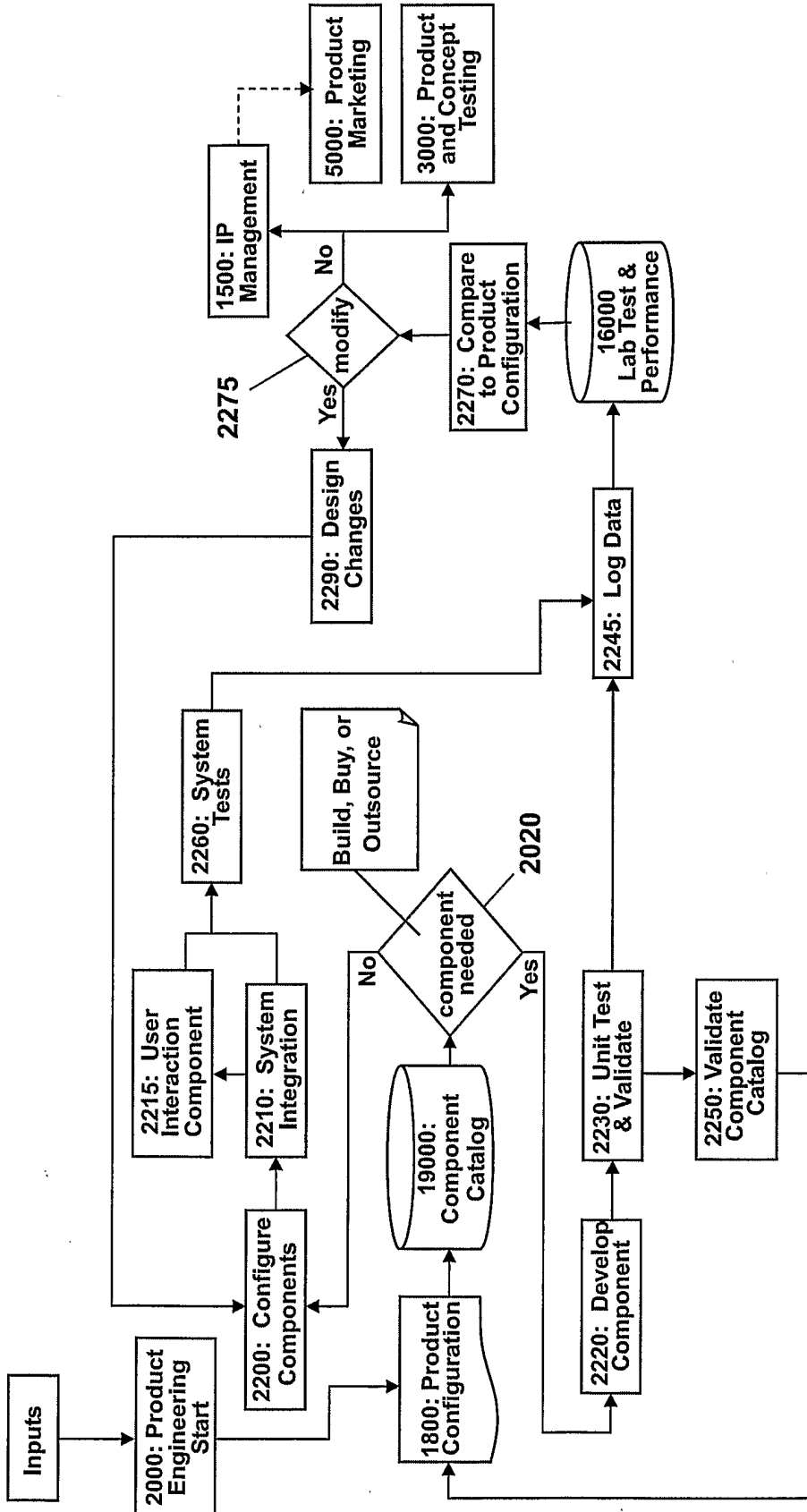


Fig. 3

5/28

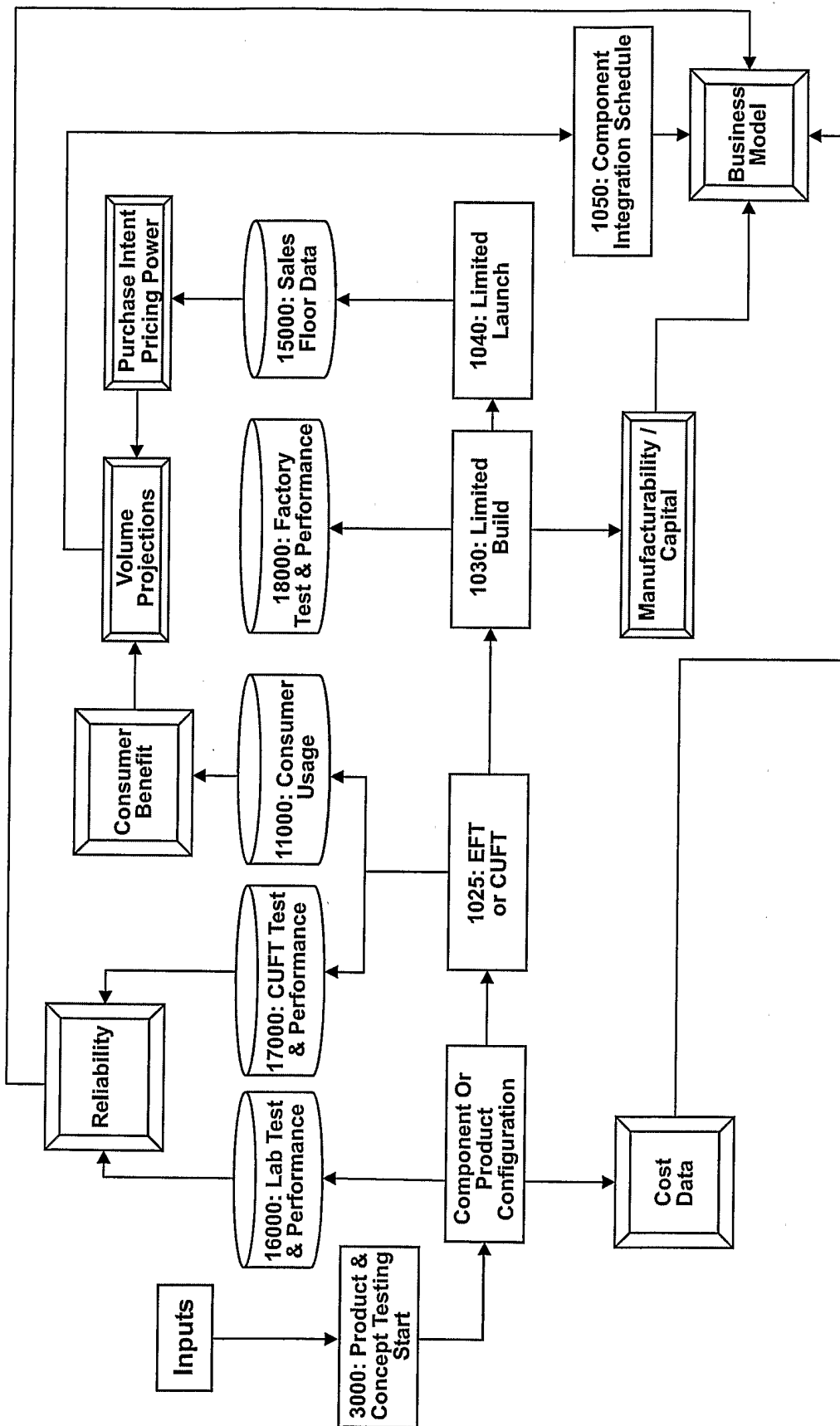


Fig. 4

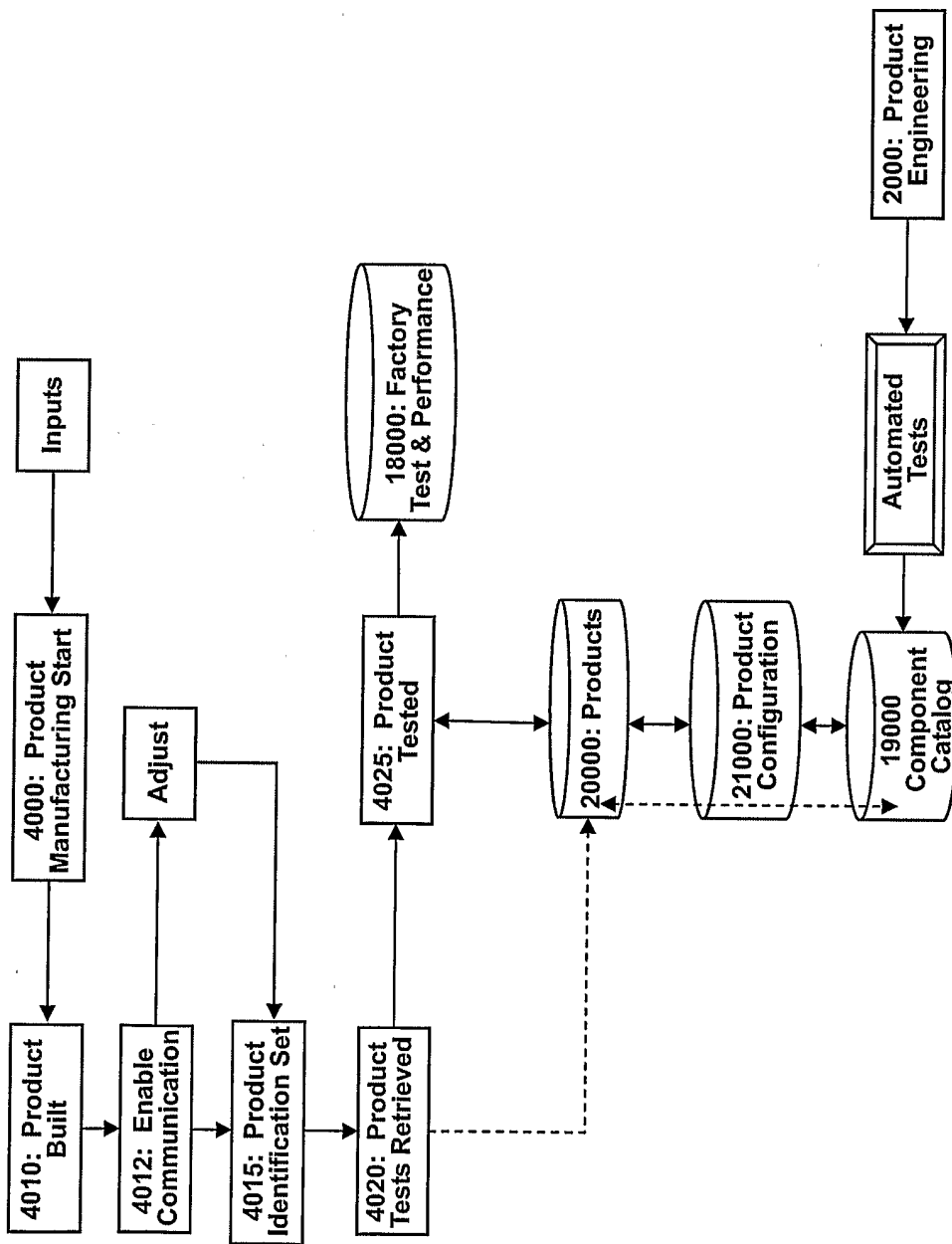


Fig. 5

7/28

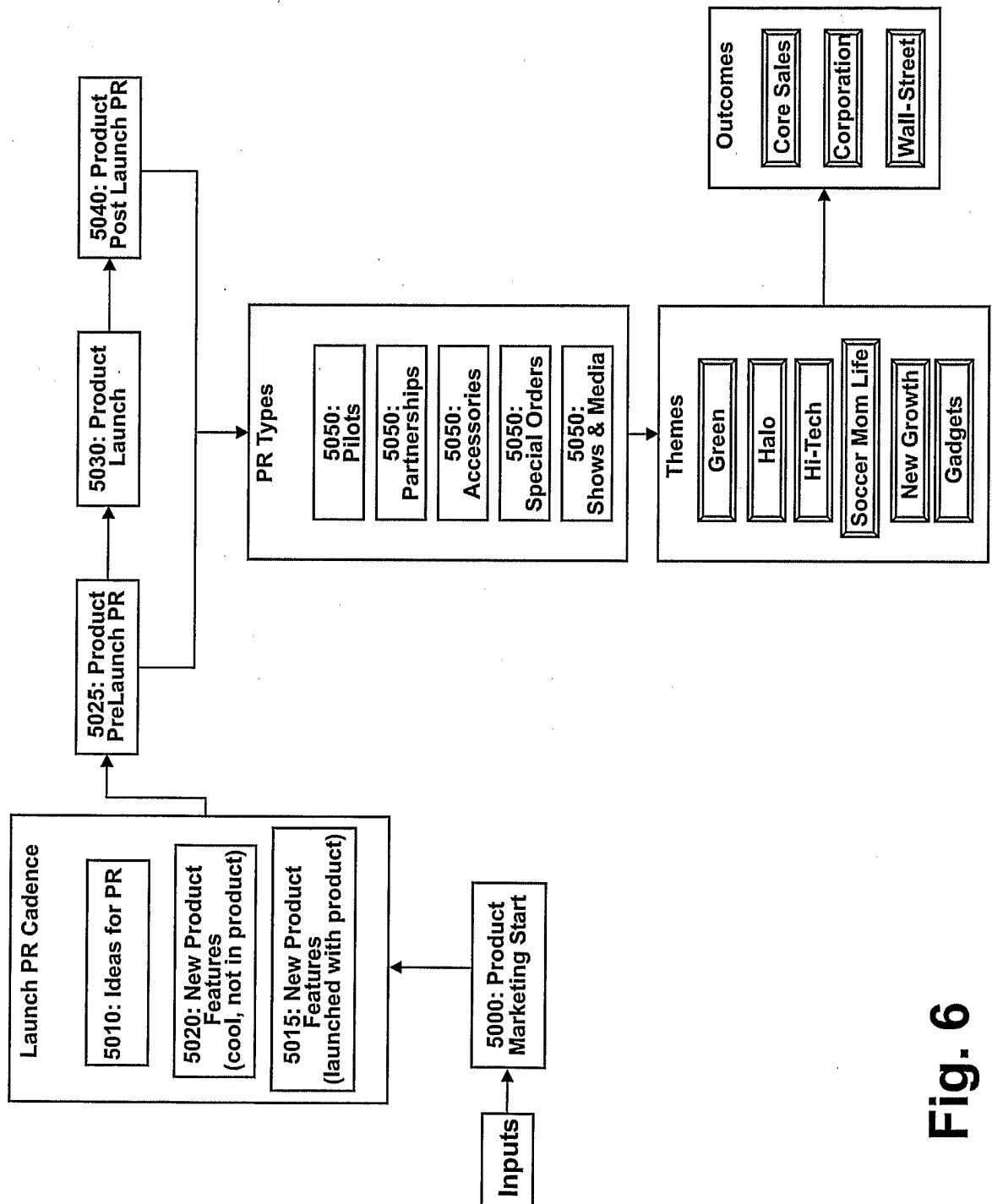


Fig. 6

8/28

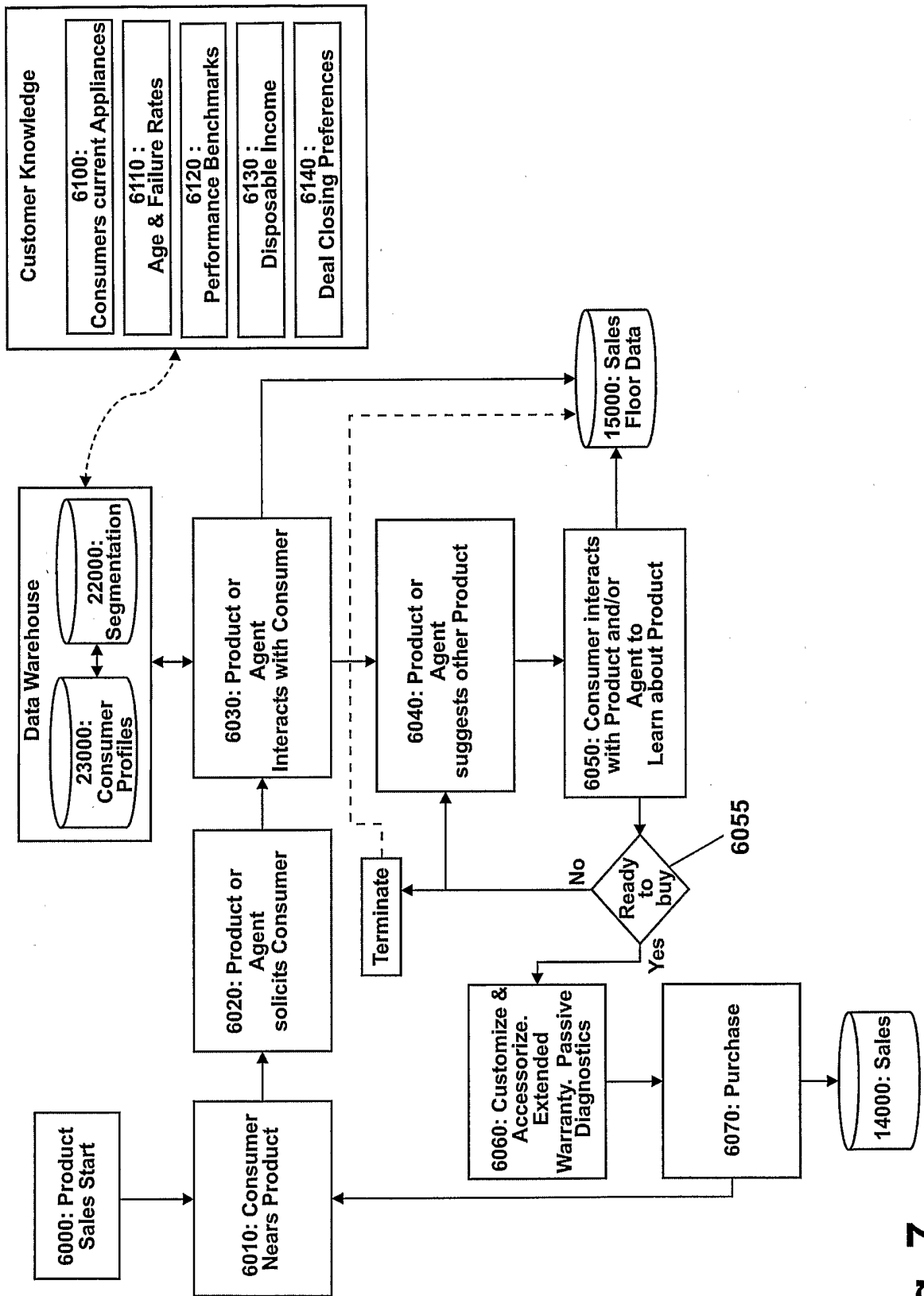


Fig. 7

9/28

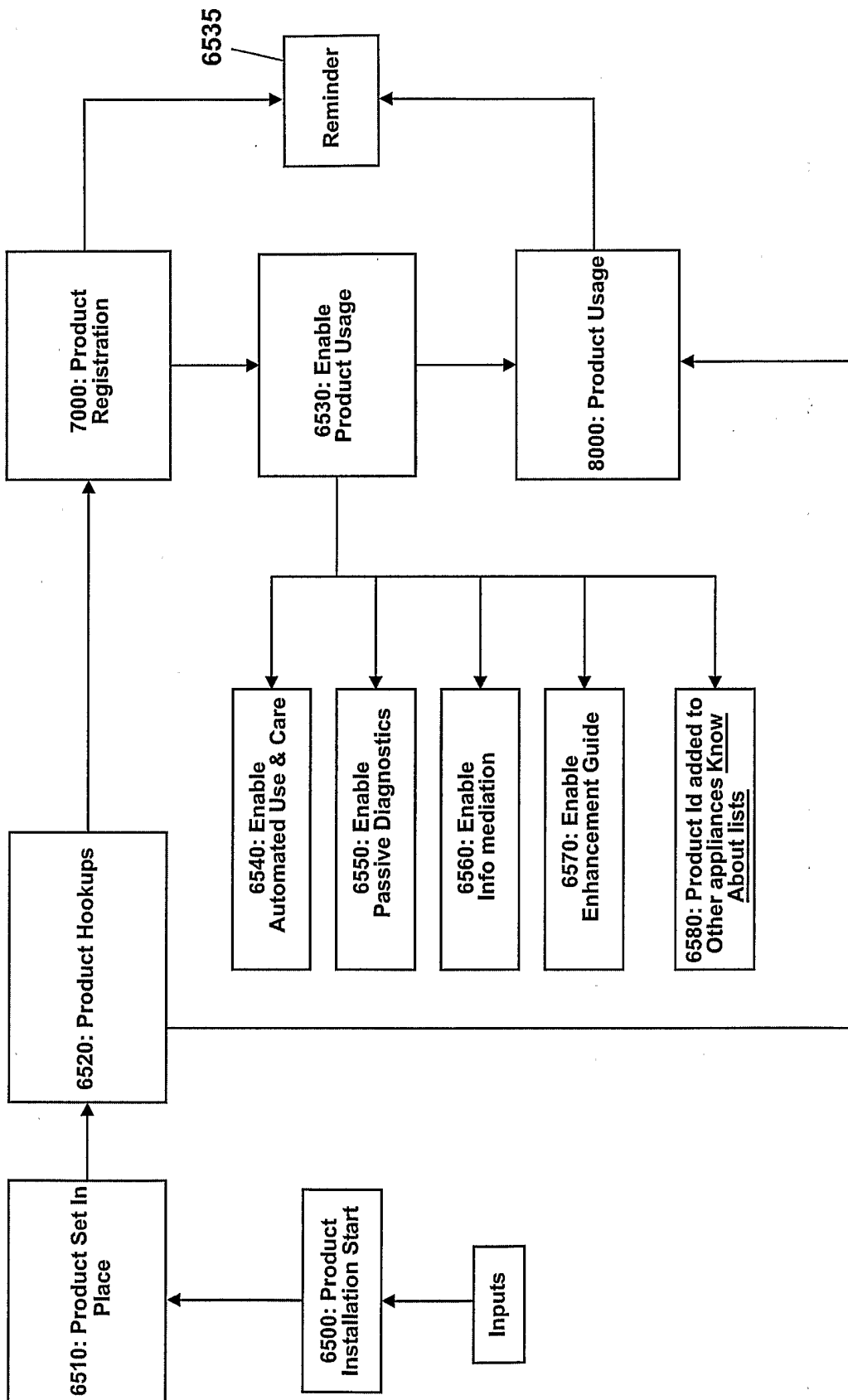


Fig. 8

10/28

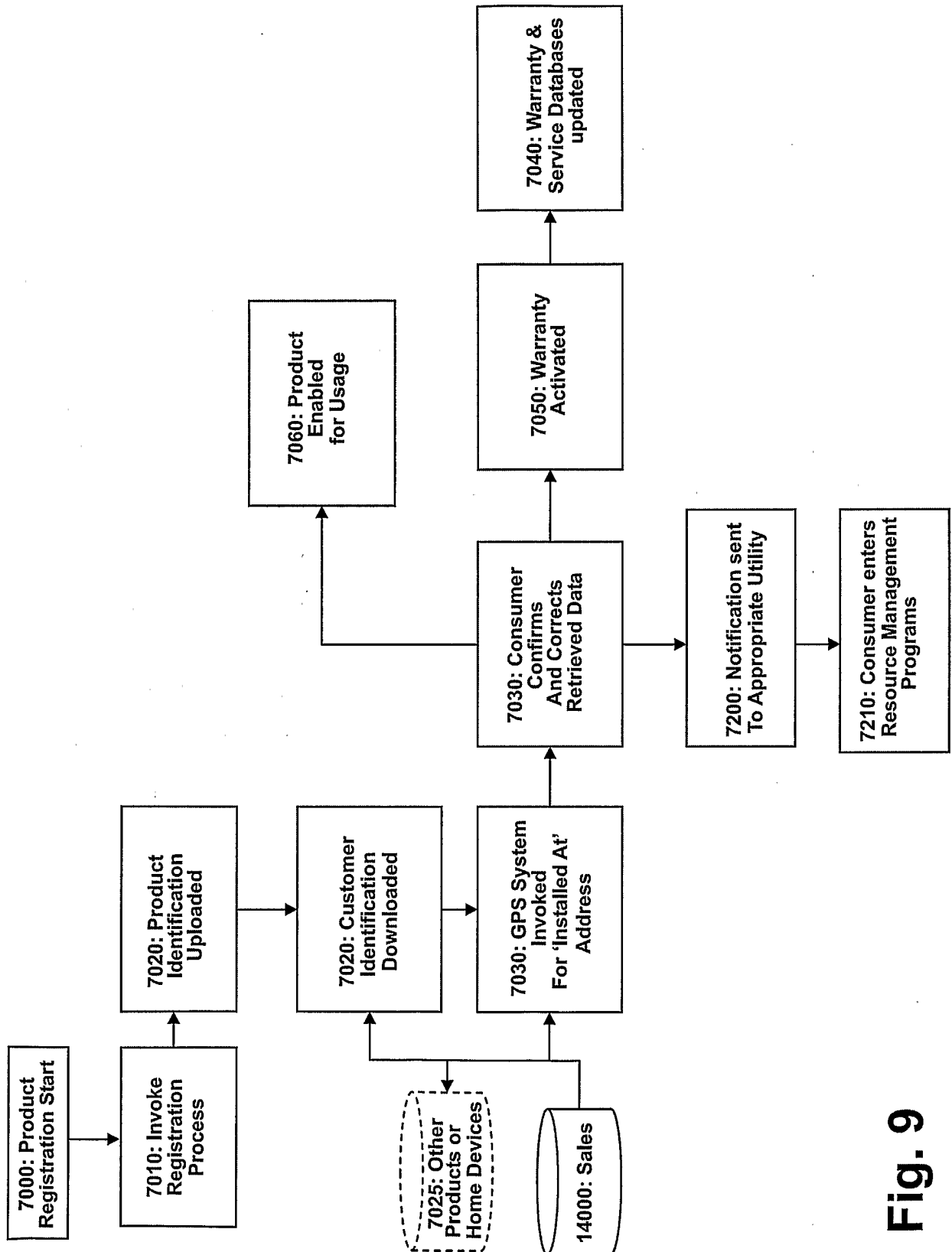


Fig. 9

11/28

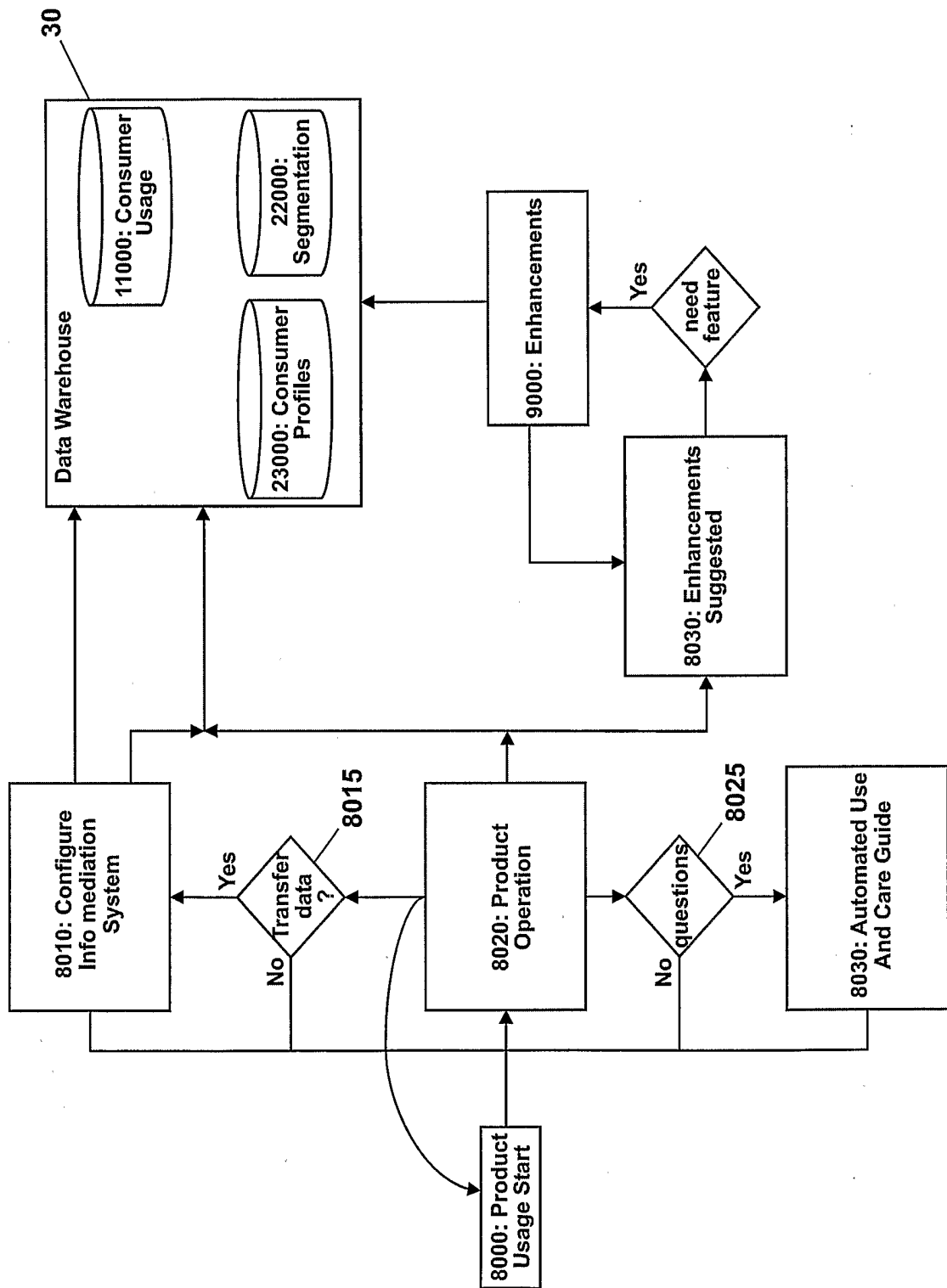


Fig. 10

12/28

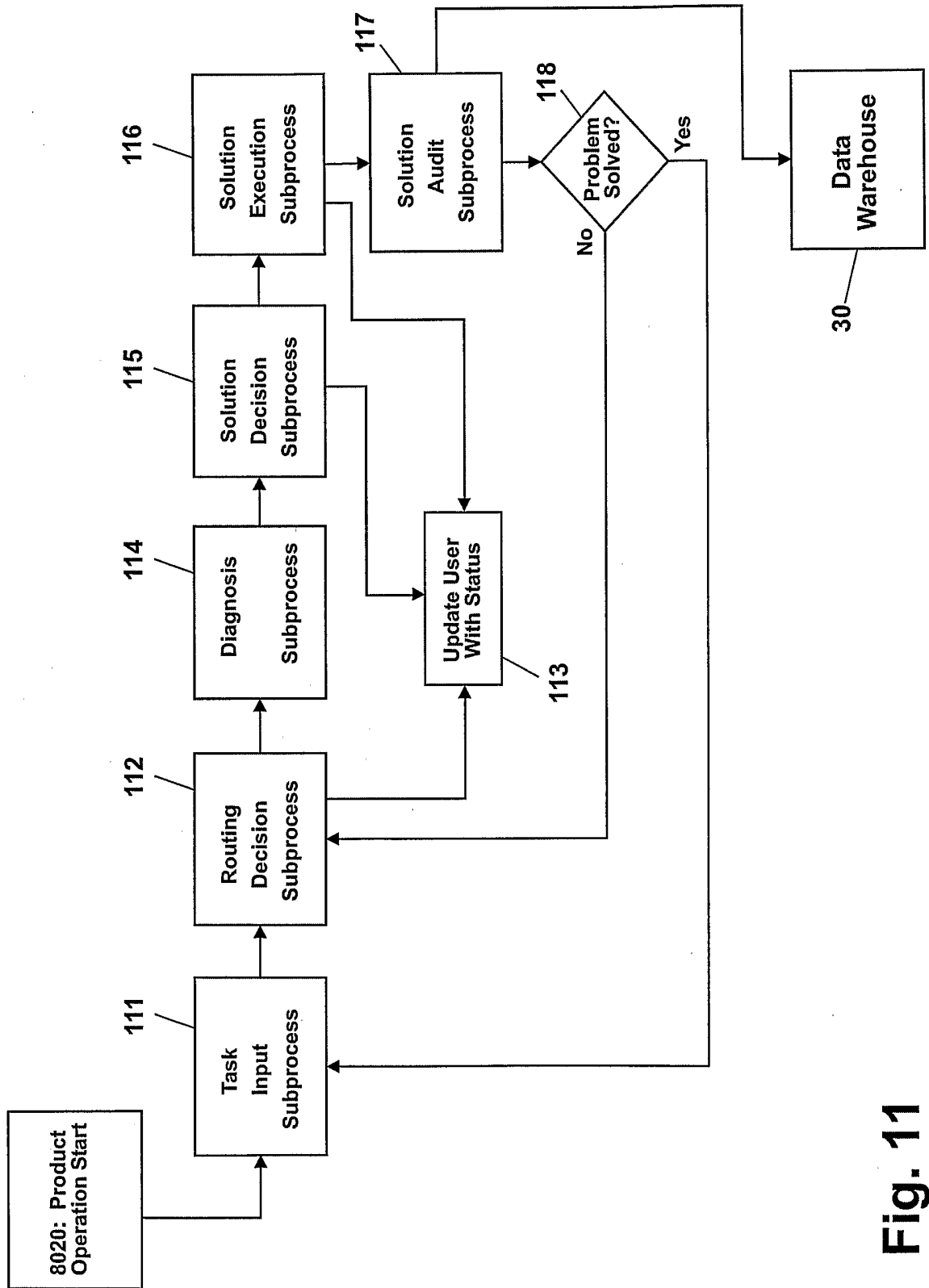


Fig. 11

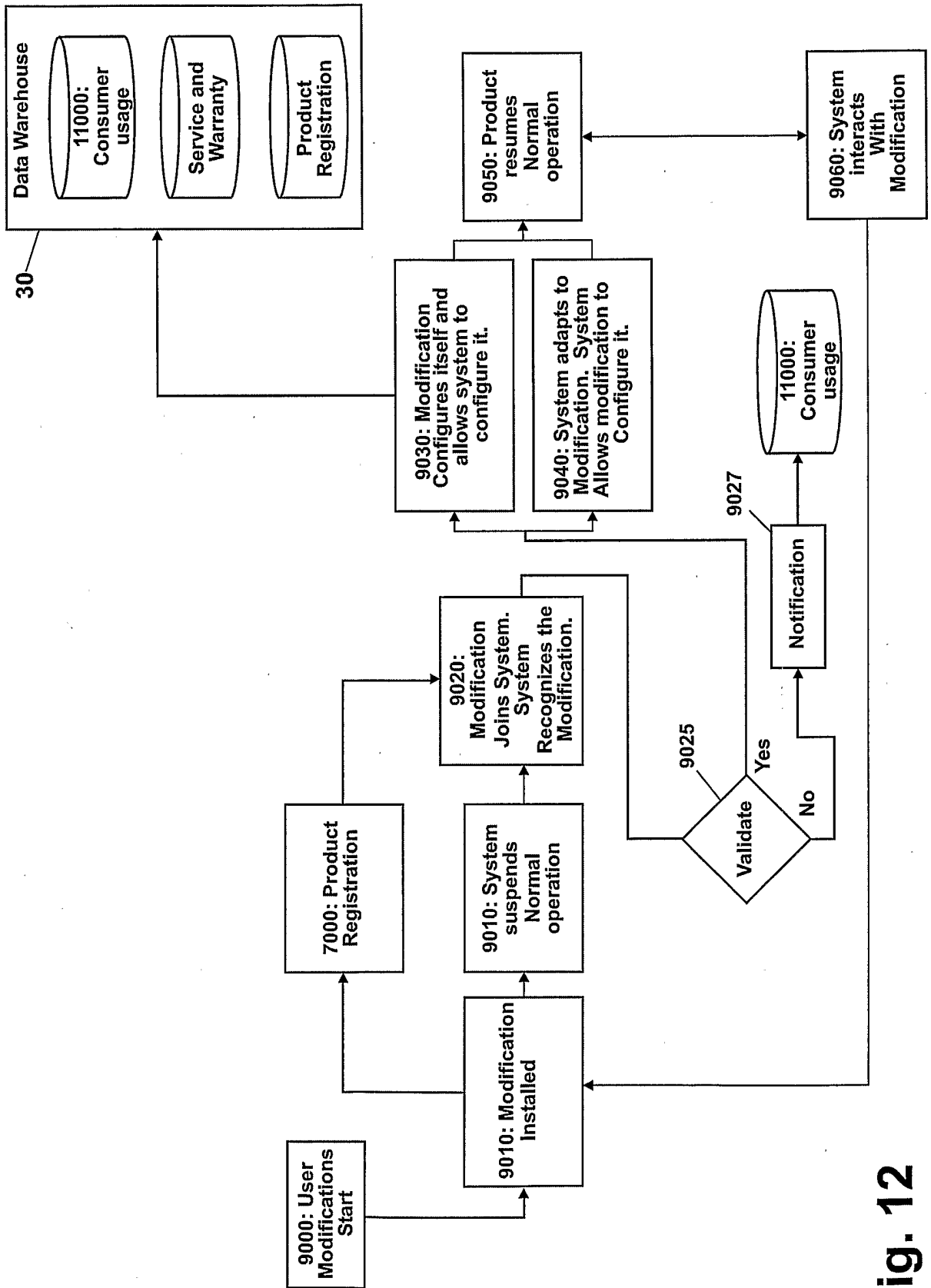


Fig. 12

14/28

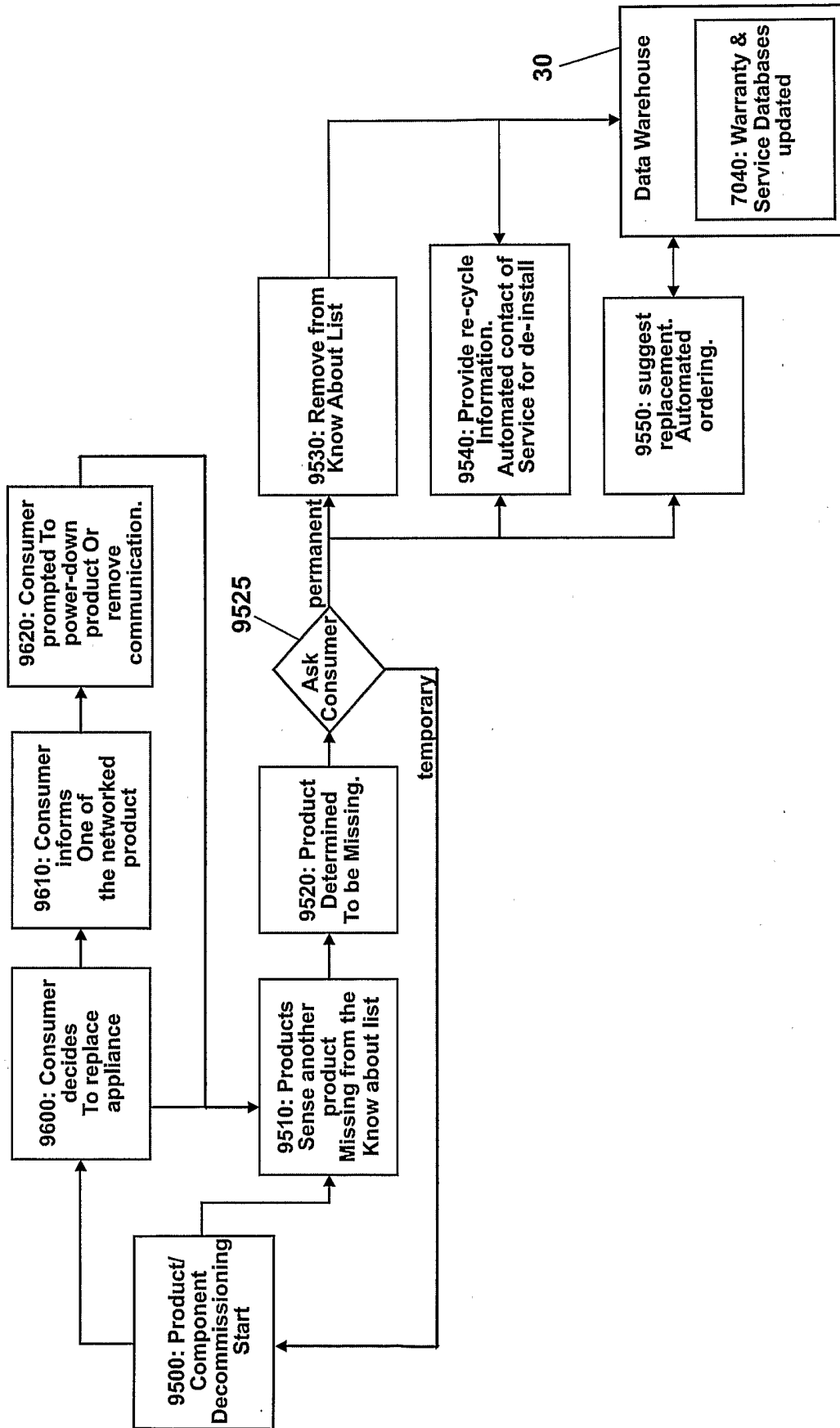


Fig. 13

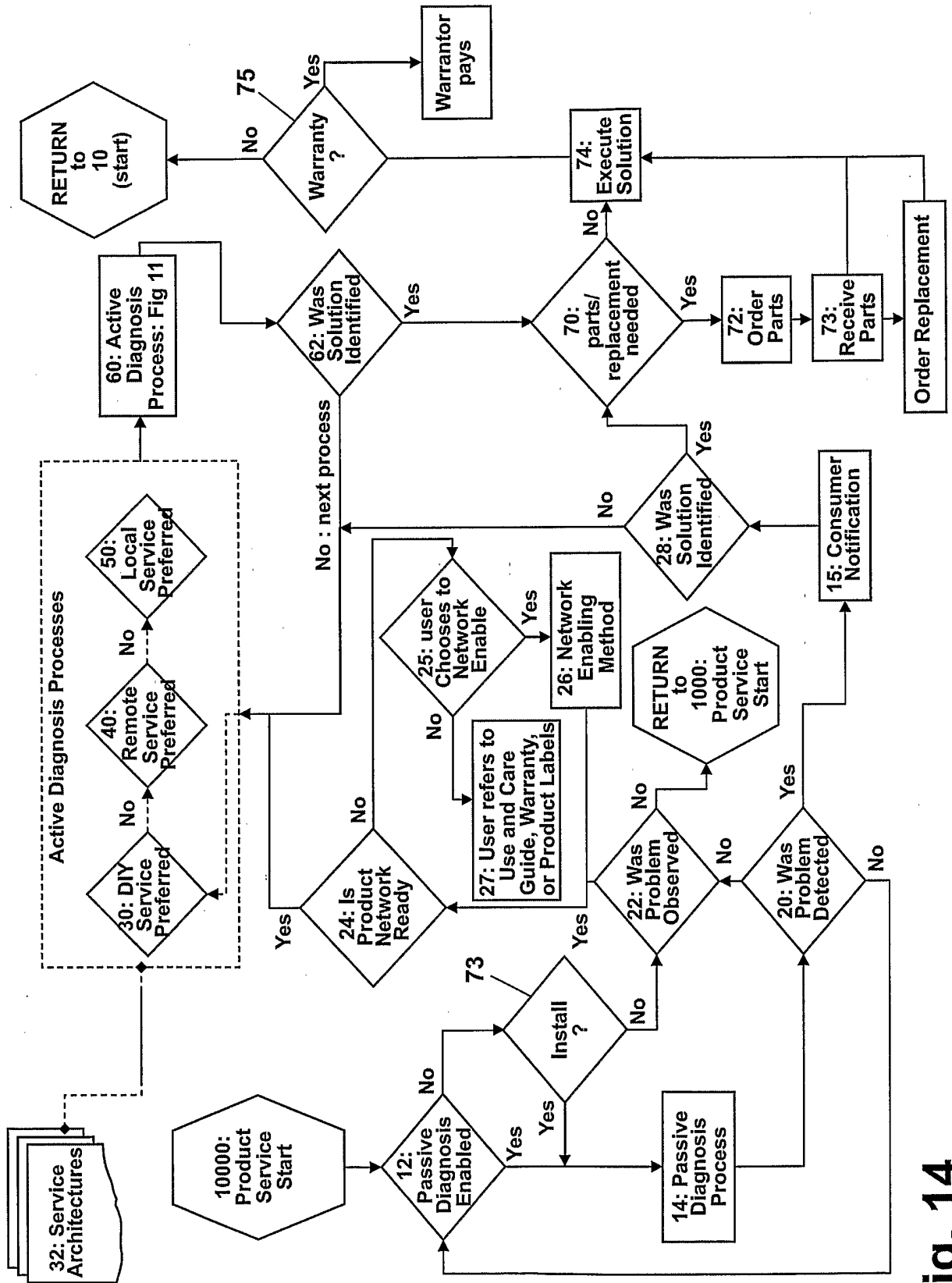


Fig. 14

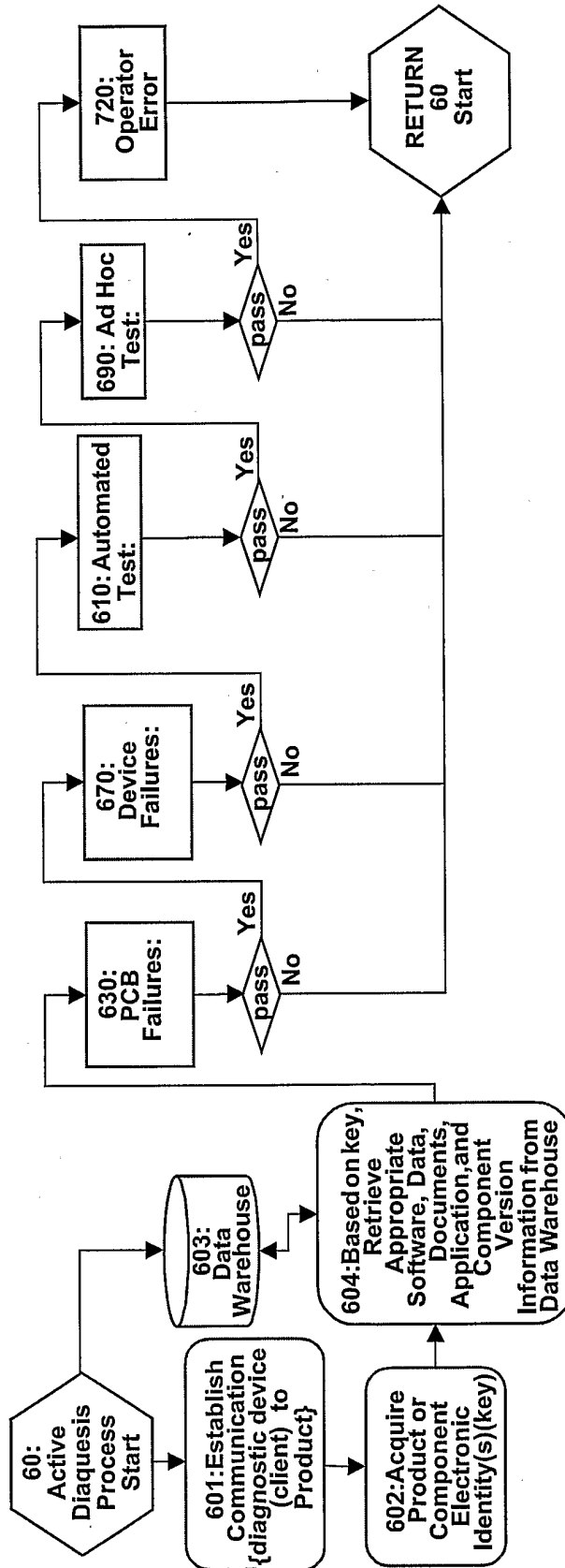


Fig. 15

17/28

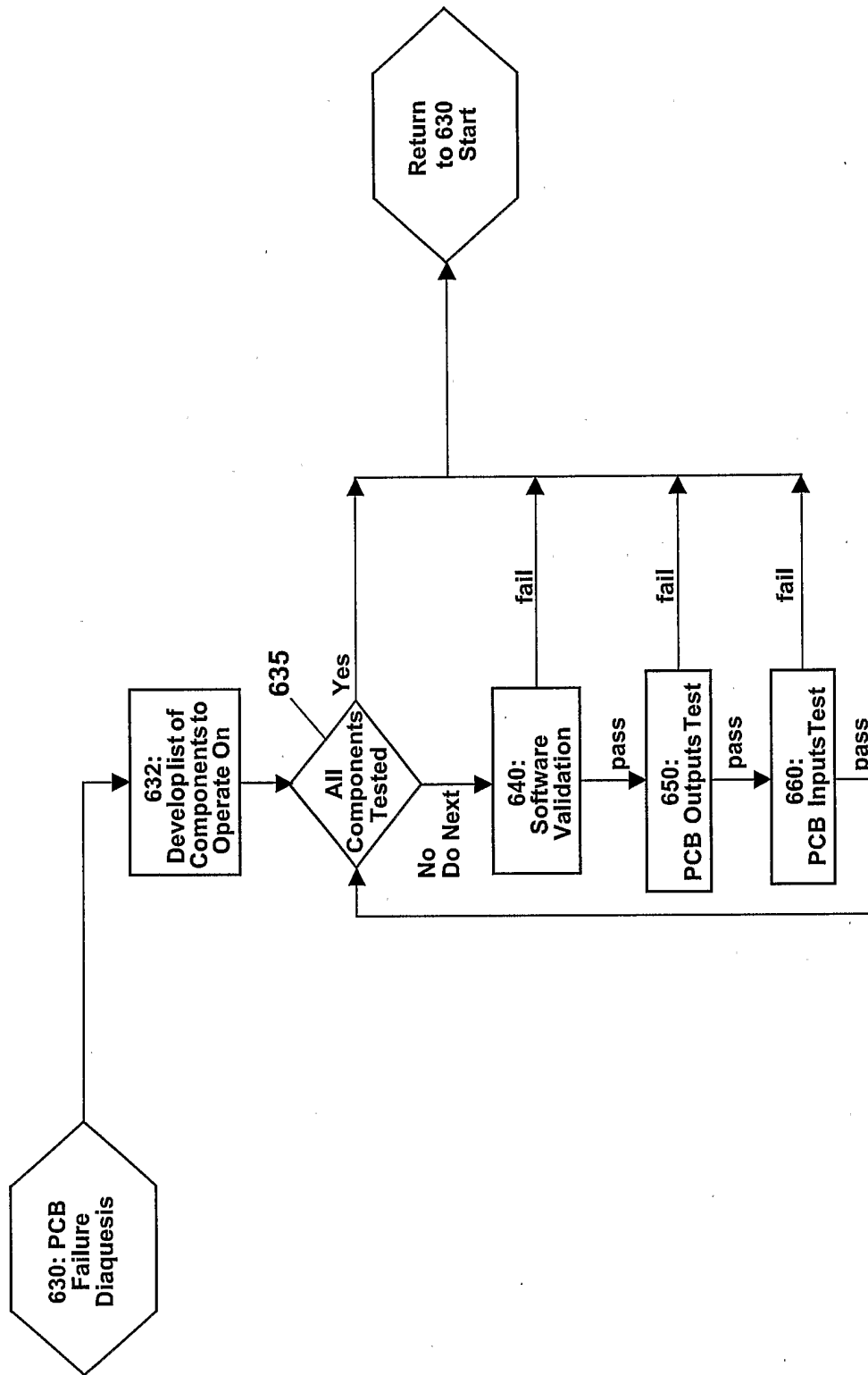


Fig. 16

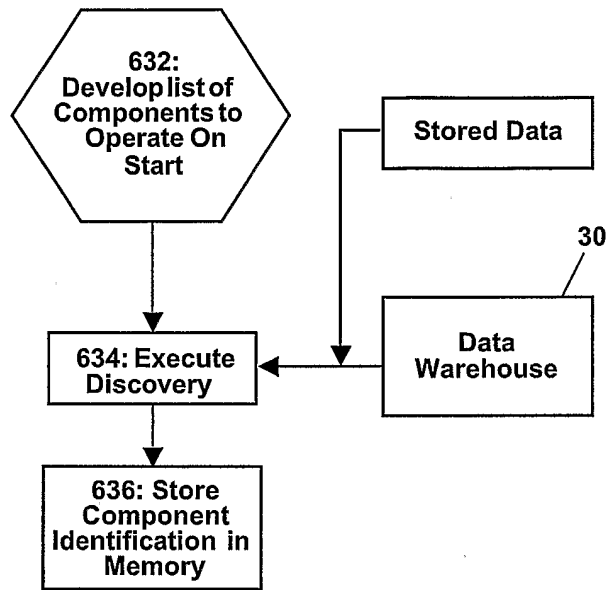


Fig. 17

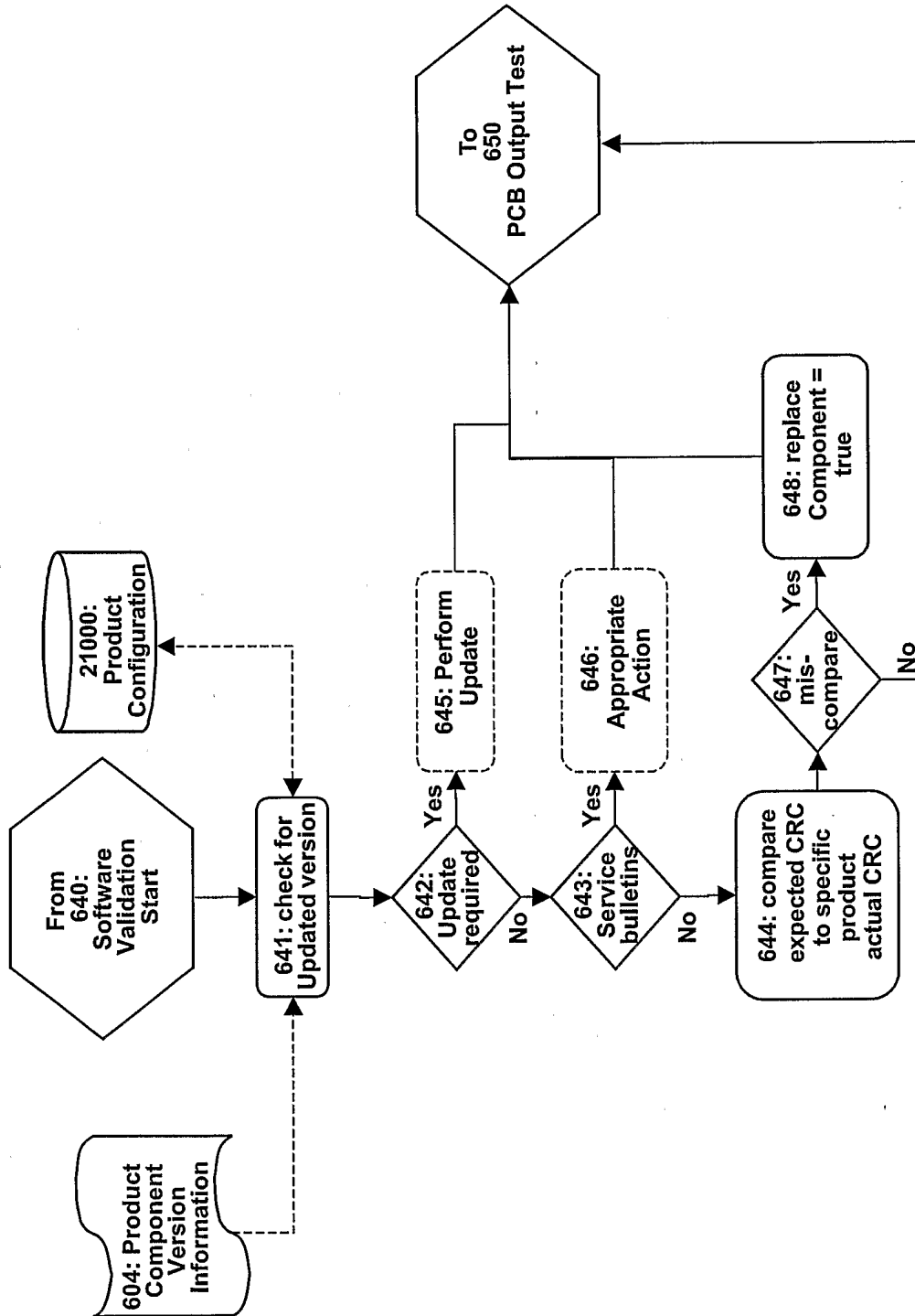


Fig. 18

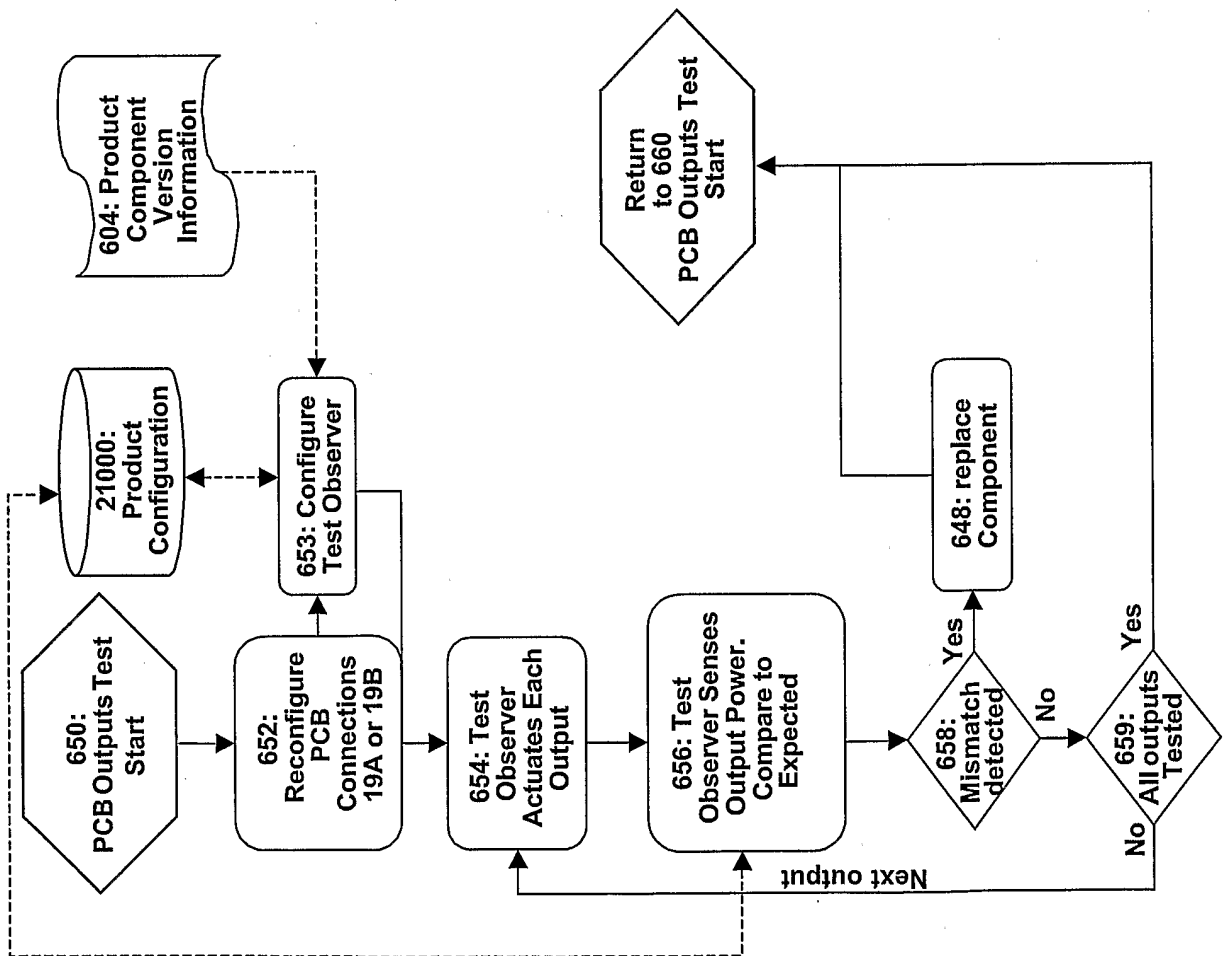
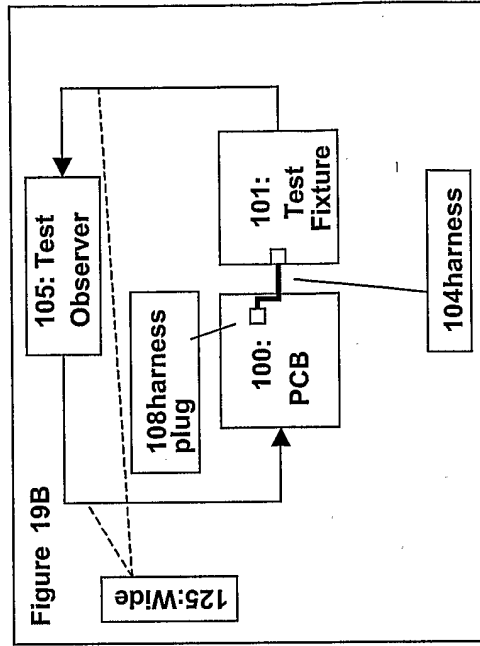
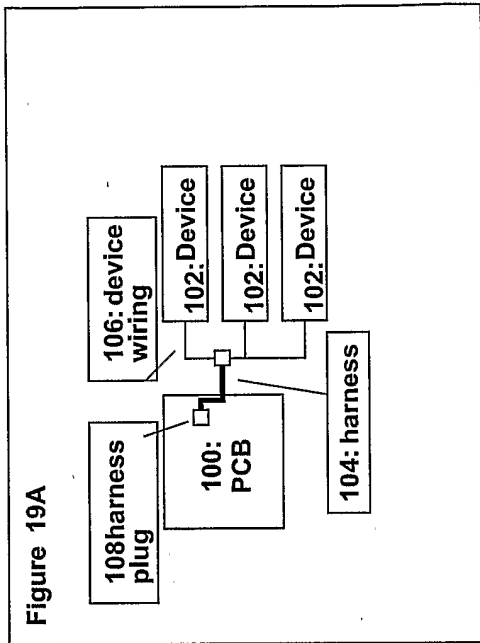


Fig. 19

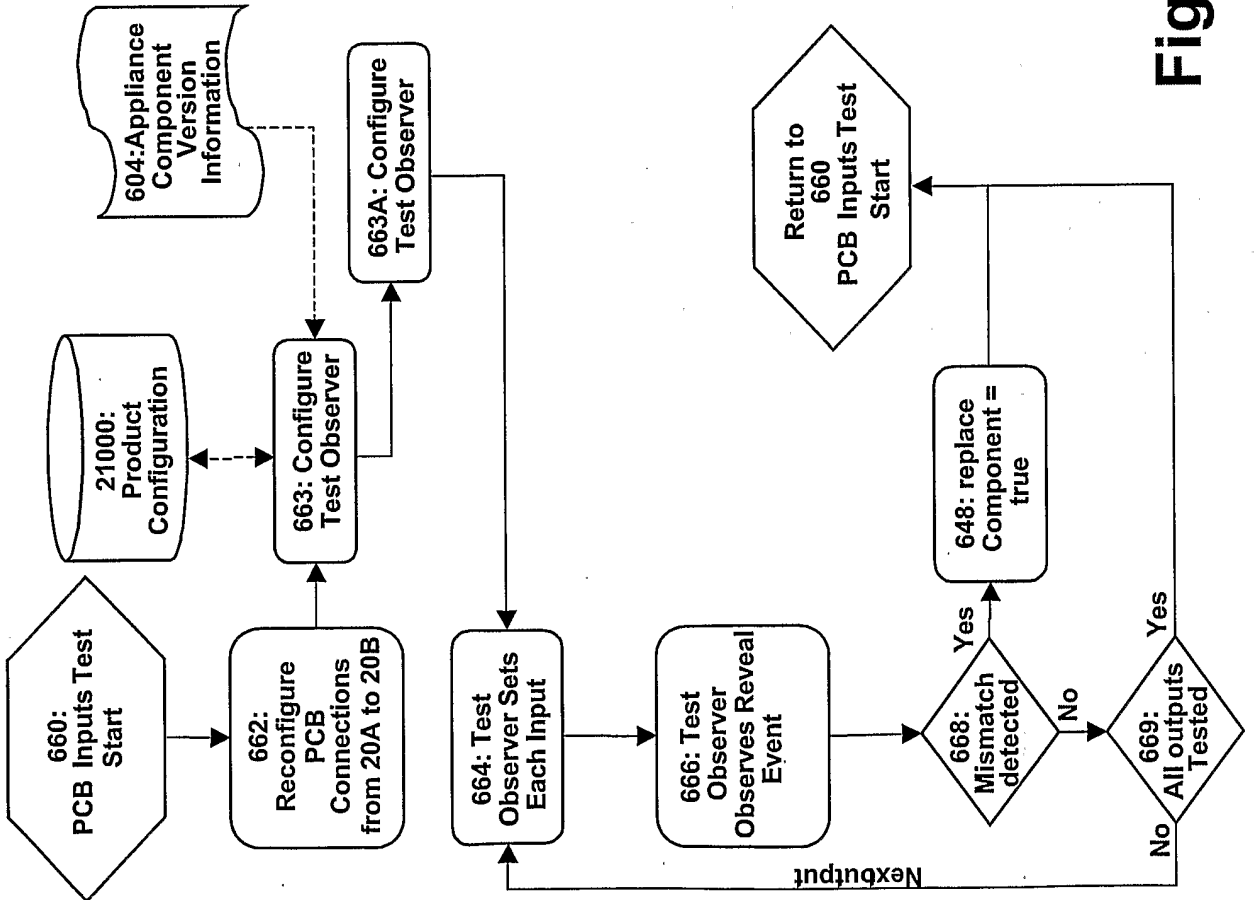
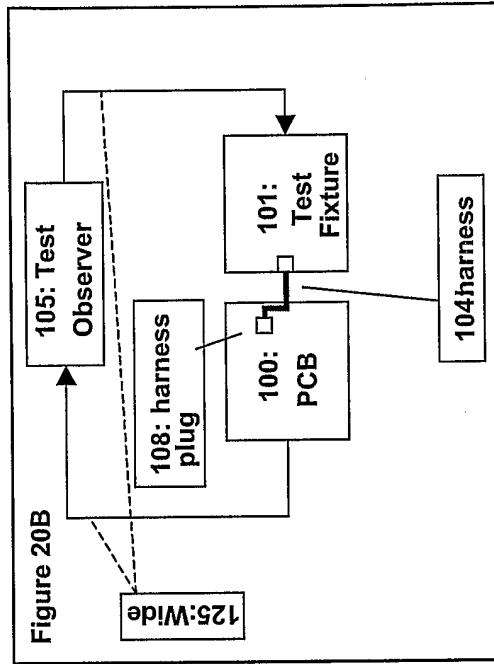
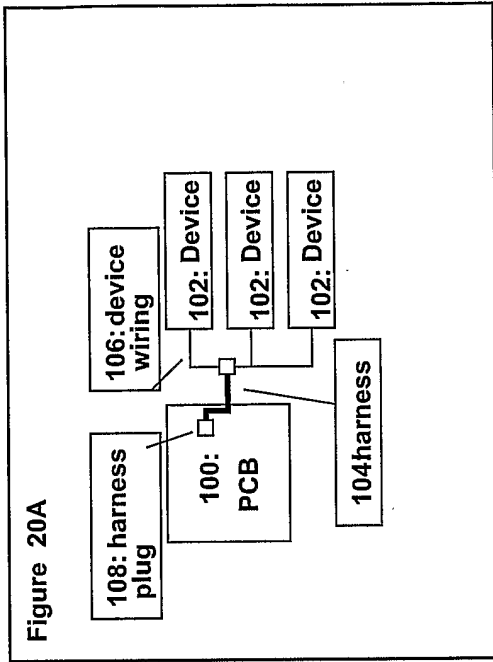


Fig. 20

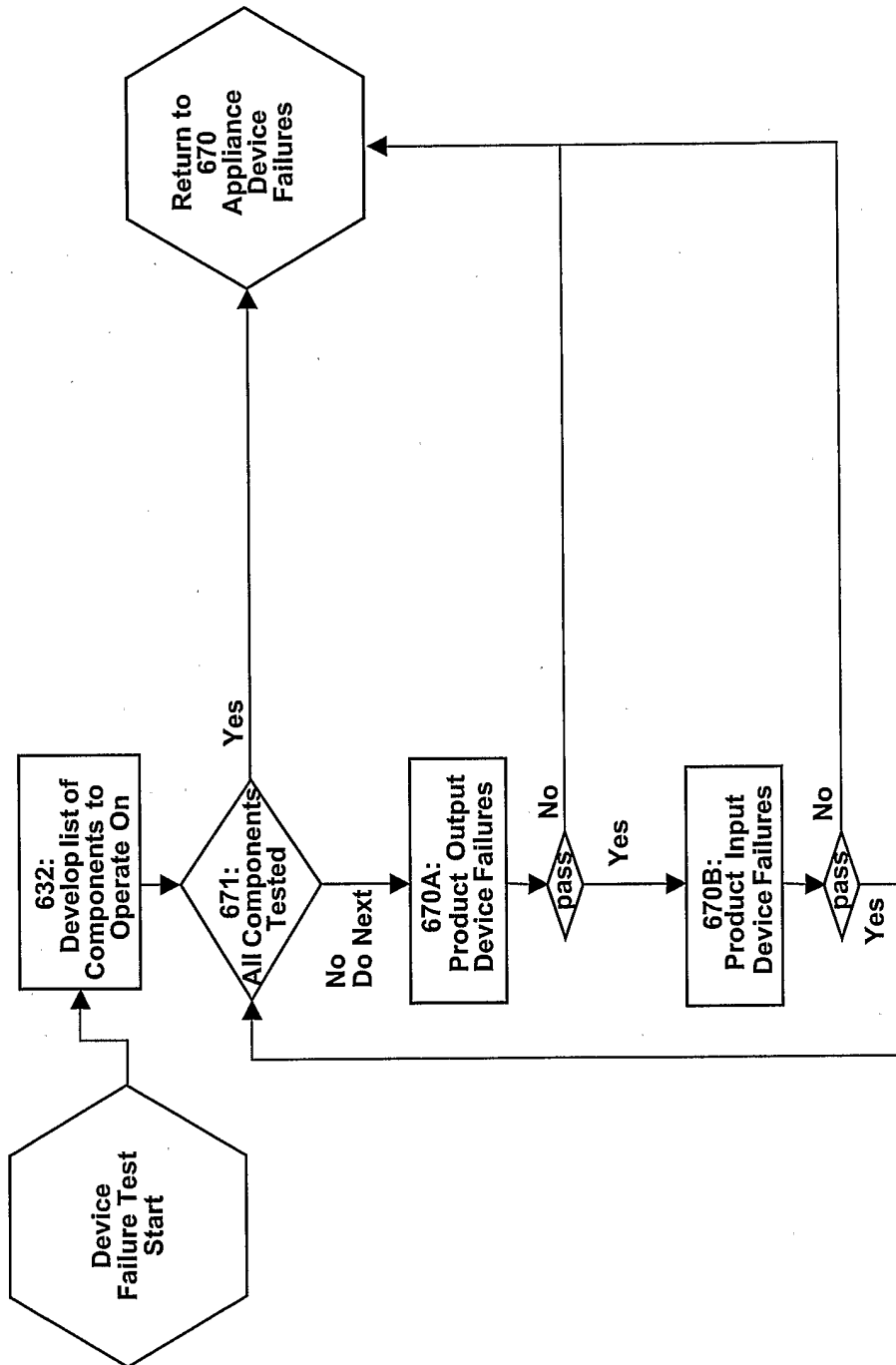


Fig. 21

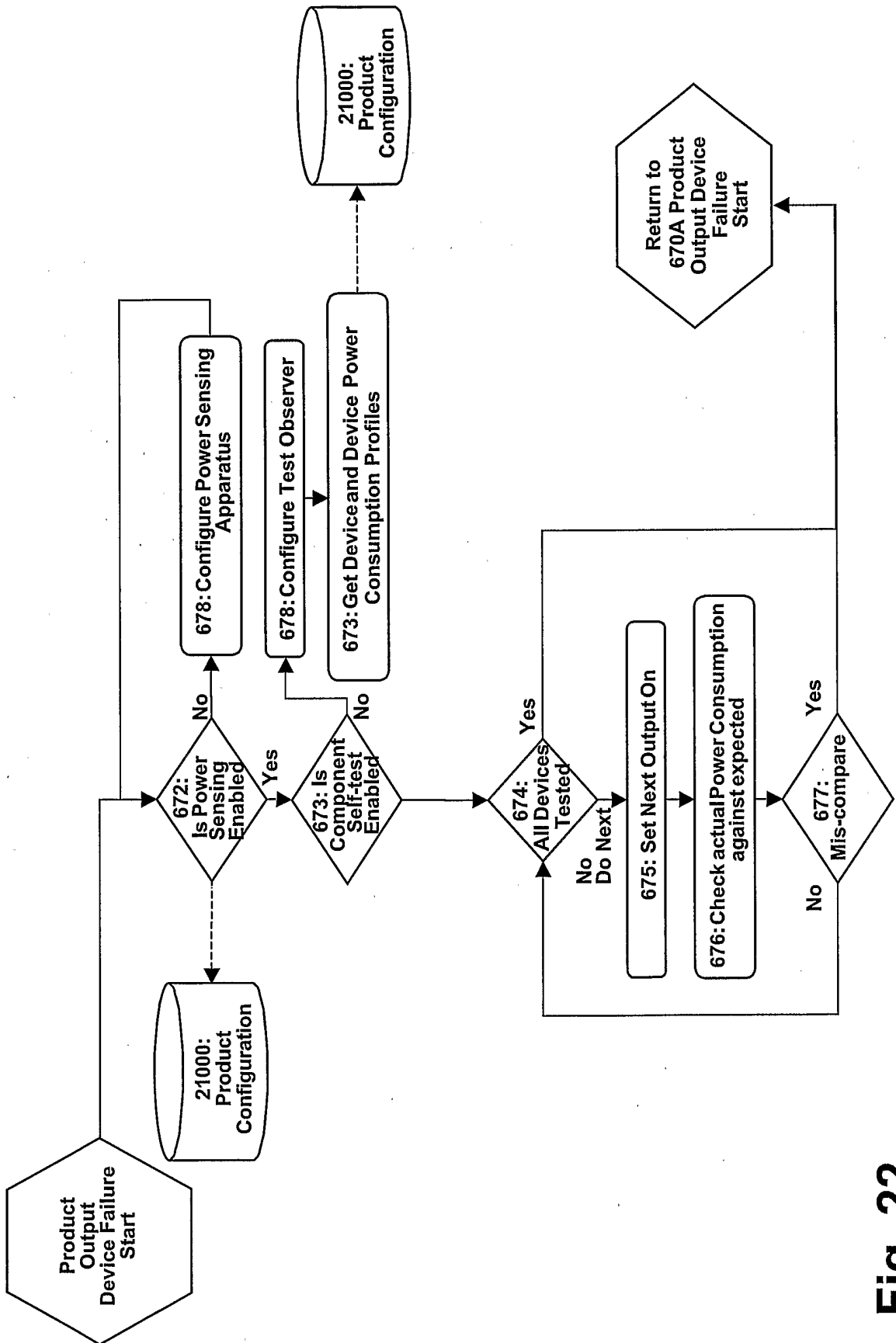


Fig. 22

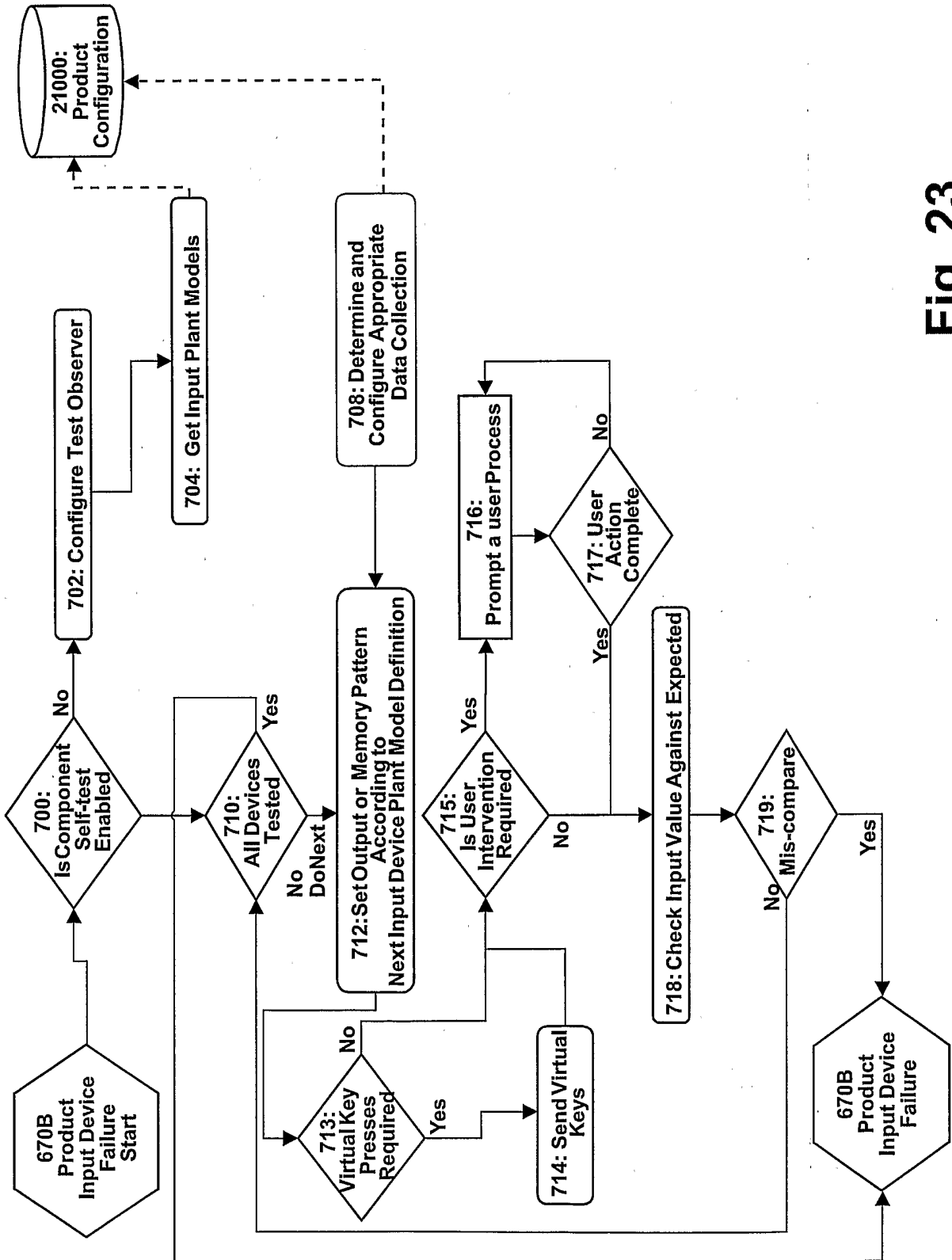


Fig. 23

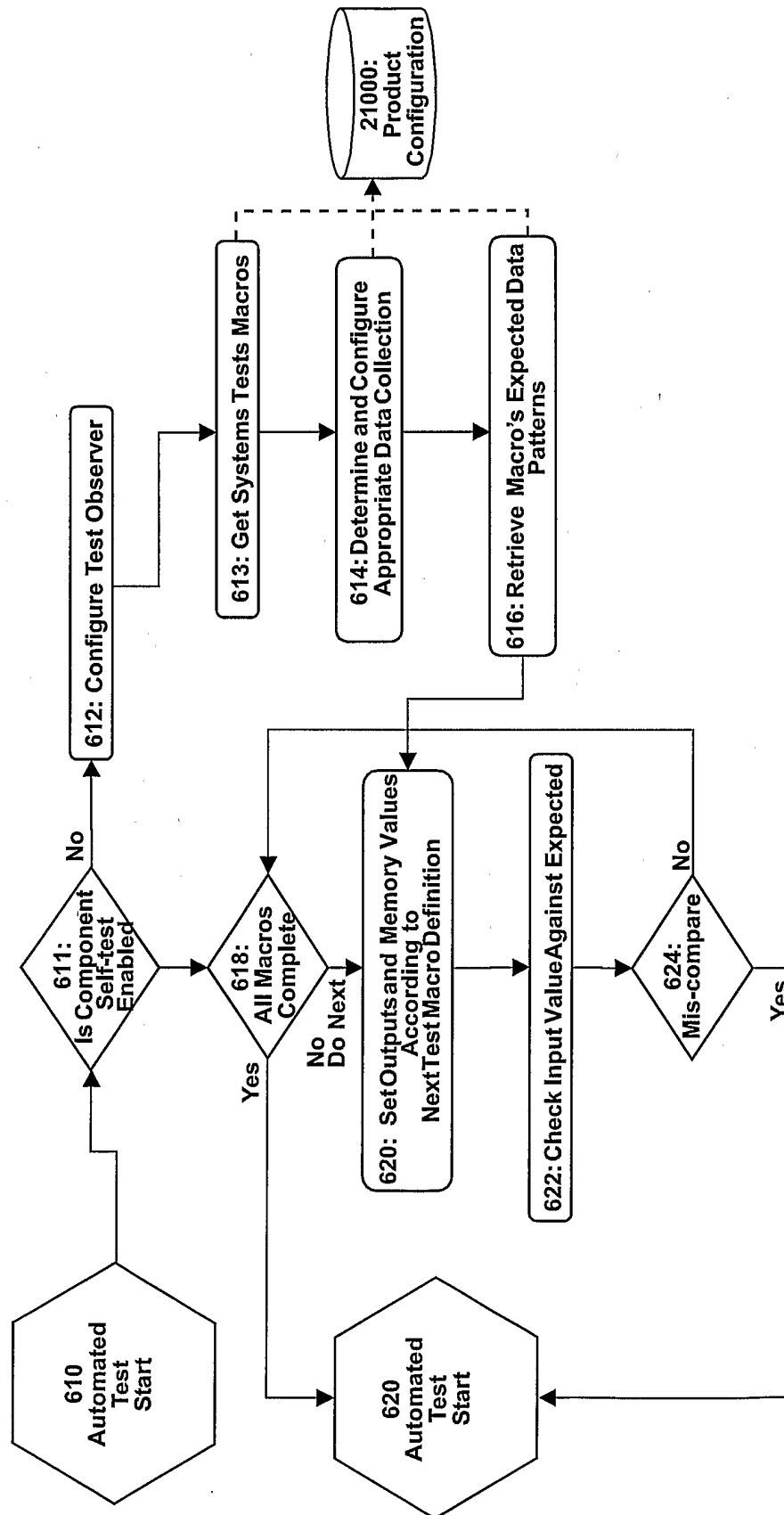


Fig. 24

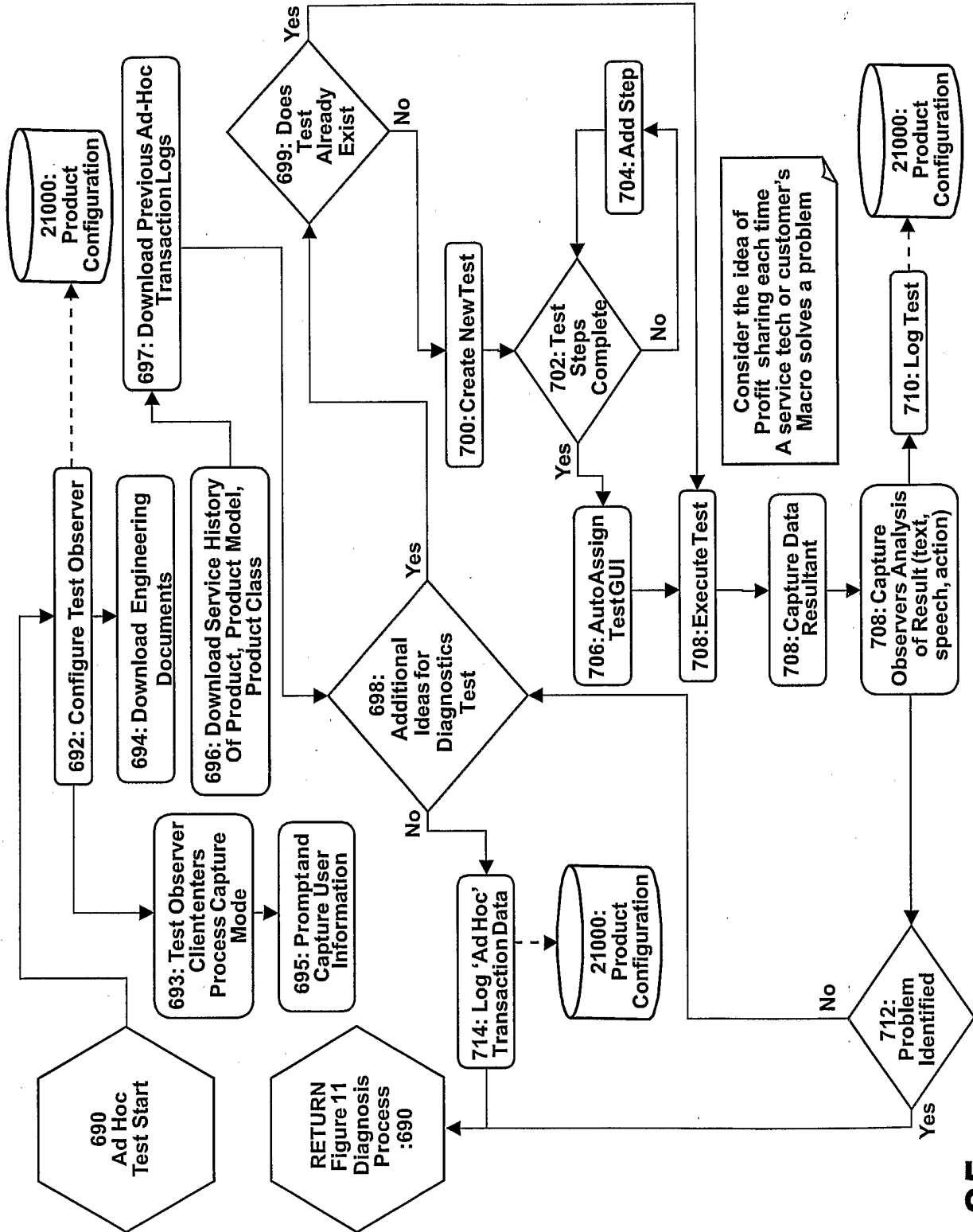


Fig. 25

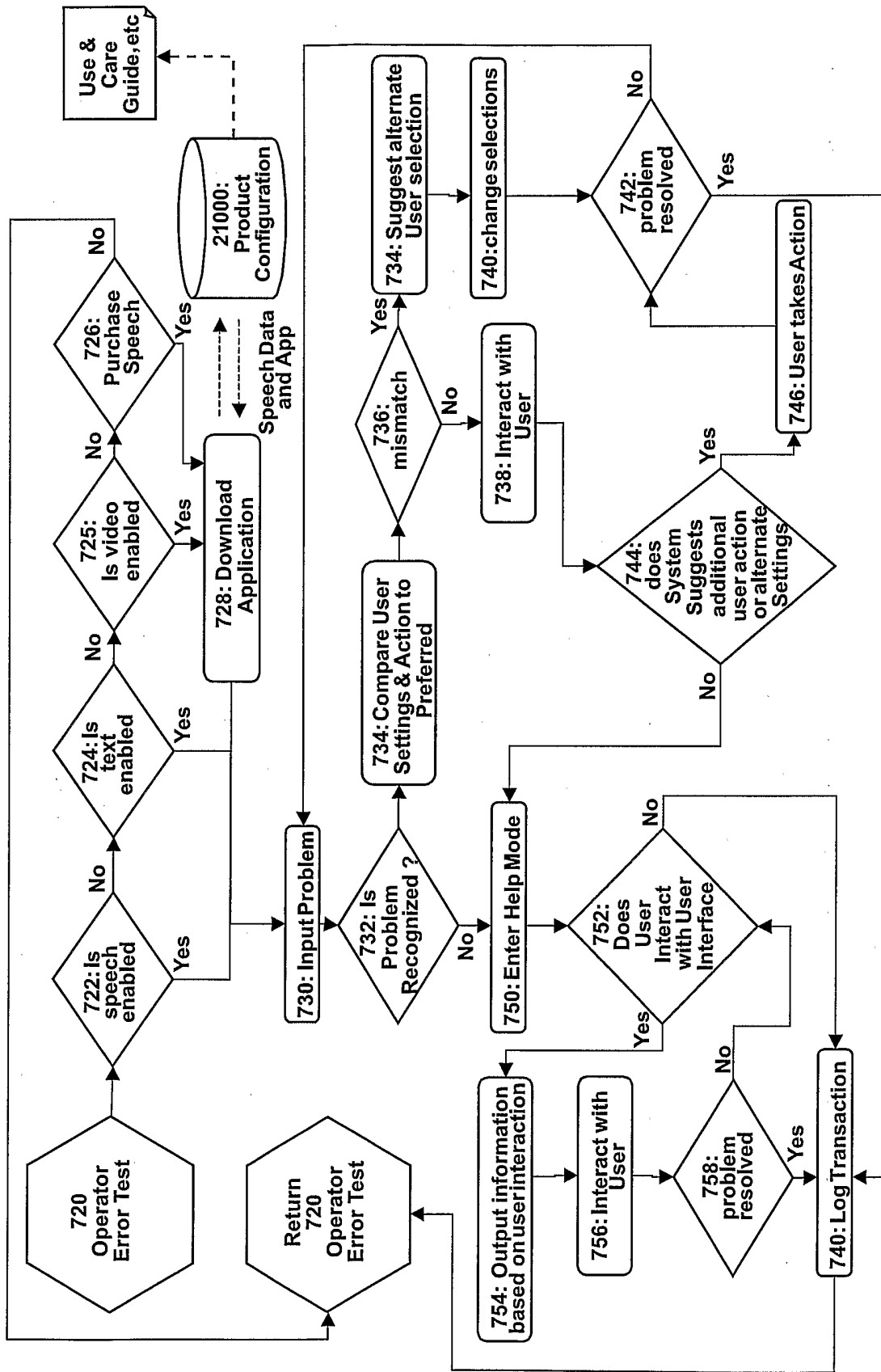


Fig. 26

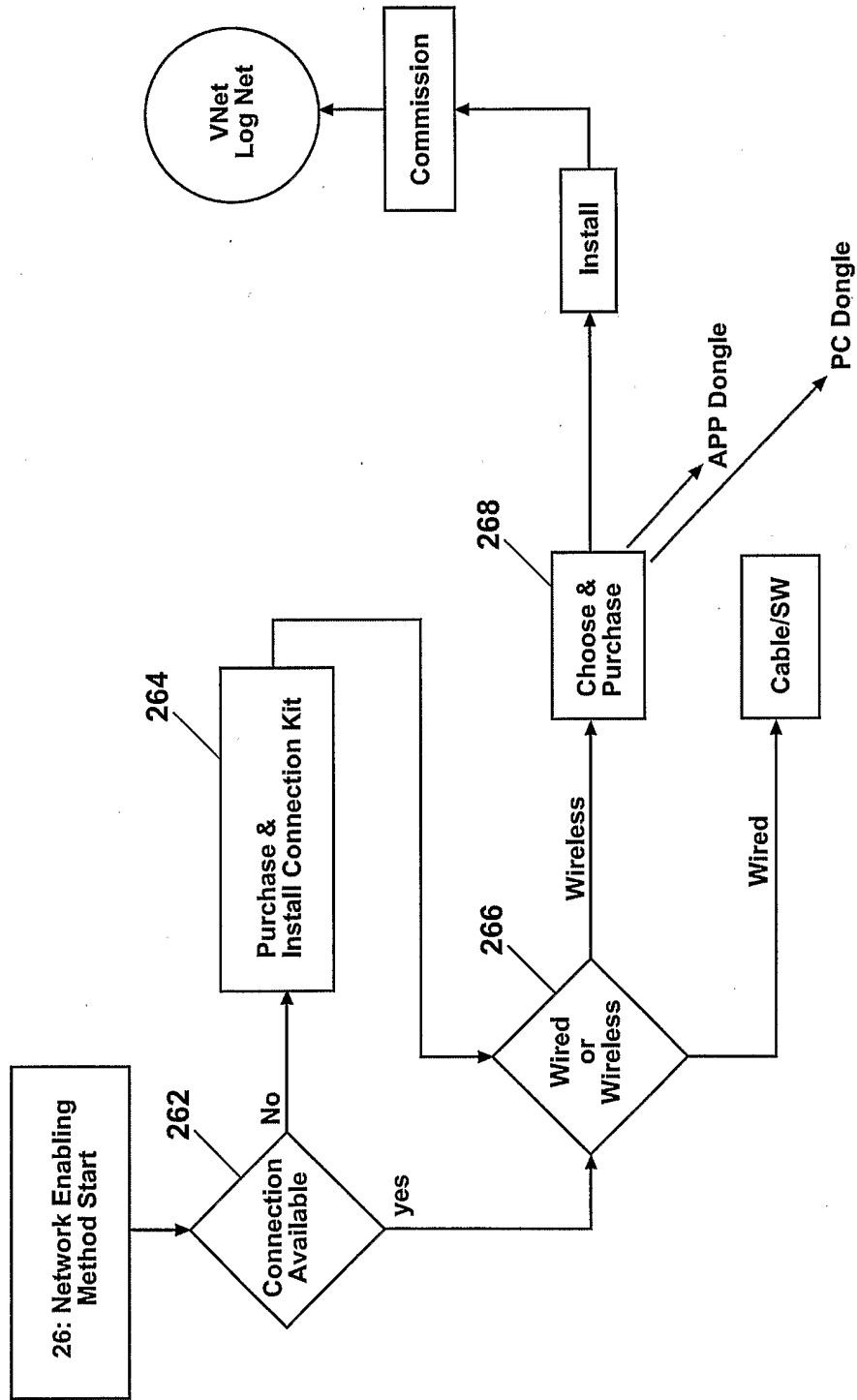


Fig. 27