

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
24 December 2003 (24.12.2003)

PCT

(10) International Publication Number
WO 03/107578 A2

- (51) International Patent Classification⁷: **H04L**
- (21) International Application Number: PCT/US03/19167
- (22) International Filing Date: 17 June 2003 (17.06.2003)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
10/173,145 17 June 2002 (17.06.2002) US
- (71) Applicant: **ADAPTIK CORPORATION** [US/US]; 630 Fairmont Avenue, Westfield, NJ 07090 (US).
- (72) Inventors: **ULUAKAR, Tamer**; 11 Tennyson Drive, Plainsboro, NJ 08536 (US). **PETTIT, John, Bartholomew, III**; 630 Fairmont Avenue, Westfield, NJ 07090 (US). **BATAKCI, Lutfu**; 1205 Pine Ford Drive,

Middletown, PA 17057 (US). **KIVANC, Bulent**; 3625 Brodhead Road, Bethlehem, PA 18020 (US). **KADAMBI, Govindaraj**; 43 Hendrickson Drive, Belle Mead, NJ 08502 (US). **MERKEL, Christopher, Fredrick**; 4412 Cheryl Drive, Bethlehem, PA 18017 (US).

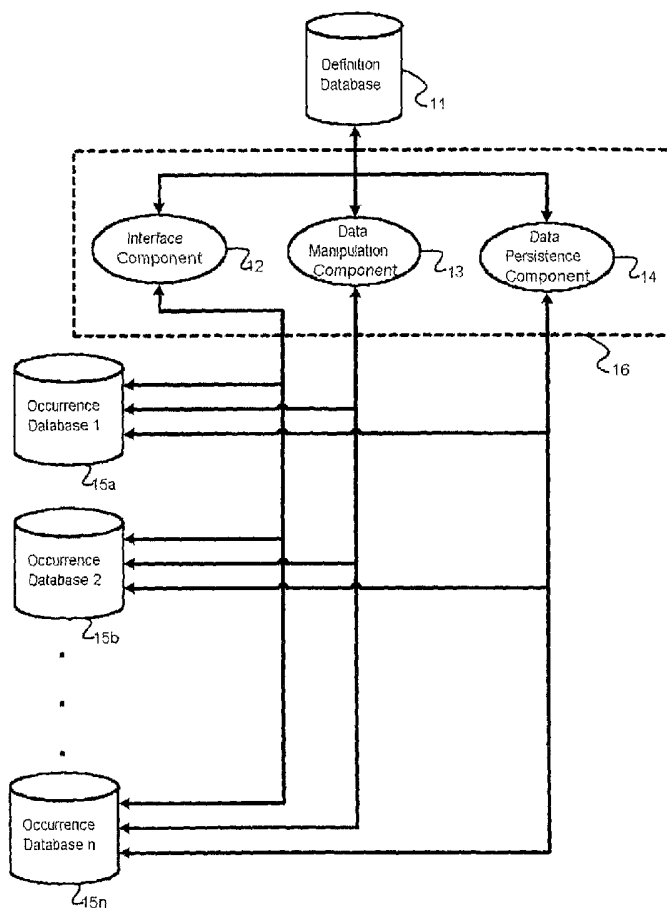
(74) Agents: **FORTKORT, Michael, P.** et al.; Mayer Fortkort & Williams, PC, 251 North Avenue West, 2nd Floor, Westfield, NJ 07090 (US).

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

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR CREATING AN ADAPTIVE APPLICATION



(57) Abstract: An Adaptive Software Application consists of several types of modules, called Adaptive Units, which are highly parameterized such that they can adapt to varying business requirements by virtue of externally provided parameters. An Adaptive Application is assembled through repeated use of various combinations of different types of Adaptive Units. Large and complex business systems can be rapidly implemented through this approach. An Adaptive Unit includes an interface component that can present information to and accept information from the outside world (such as a web page or a system interface), a processing logic component that can manipulate and evaluate information based on received parameters received (such as comparisons and decisions as in the case of data dependency decisions), and a data persistence logic component that retrieves, adds, updates, and deletes data targeting one or more Occurrence Databases. All three components of an Adaptive Unit are parameter driven. These parameters are not specific to any particular business. One embodiment for providing parameters to these components includes a Definition Database. There may be one or more Occurrence Databases depending on the number of business processes supported by the business application for which the Adaptive Application is being adapted. The Occurrence Database includes generic (also referred to as meta model driven) tables that are not specific to any particular business.



Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,
ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,
SE, SI, SK, TR), OAPI patent (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*

**METHOD AND APPARATUS FOR
CREATING AN ADAPTIVE APPLICATION**

FIELD OF THE INVENTION

[0001] The present invention relates generally to methods and apparatuses for developing software, and more particularly to a method and apparatus for developing software for business solutions in a rapid manner.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material that may be subject to copyright protection. The copyright owner has no objection to the facsimile reproduction of the patent document or the patent disclosure, as it appears in the U.S. Patent & Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND

[0003] To date, software applications for businesses have required significant development time and once built cannot be easily modified to adapt to changes in business needs. Given the length of the development cycle vis-à-vis the length of the business cycle, this almost always leads to software systems that are outdated the moment they are completed.

[0004] For example, in the insurance industry computer systems exist to administer insurance policies for large insurers. These large software systems took years to implement and are not flexible enough to evolve with the business. As a result, most insurance companies today have out-of-date and inflexible systems and are confronted with excessive costs and delays in introducing new technologies and maintaining legacy systems. For example, the cost of handling year 2000 issues in the insurance industry was exorbitant. Consequently, many insurance companies seek to free themselves of their legacy system overhead and replace them with web-based solutions that are easy to use, support multiple insurance products, and keep up with changes in the business.

[0005] An insurance executive who decides to address these issues and surveys similar projects recently undertaken by other insurance companies will find an inadequate value proposition. The planned cost for such a project ranges from \$10,000,000 to \$50,000,000 just for software, consulting, and personnel expenses. The planned time to complete the entire system is 2 to 4 years without any tangible result for about 18 months. The project requires input from a number of knowledgeable business staff that will typically need to be dedicated to the project for months or years. In addition, the project has a very high risk of failure. Most projects far exceed the planned cost and time. Some projects are terminated with no result. Those that are implemented often lack the functionality and flexibility initially envisioned, resulting in yet another “legacy” system.

[0006] In an attempt to deal with their past inability to address the issues of speed, cost, and flexibility, software vendors have recently been emphasizing the “componentized” nature of their software. In theory, component-based software architectures enable customers to develop systems quickly by reusing components, ultimately reducing system maintenance costs.

[0007] While the advantages of this component rationale have been highly touted, the expected benefits have yet to manifest themselves. Current projects utilizing even the newest software products marketed in this manner are still facing implementation schedules of two to four years or more. In reality, considerable programming effort is required to implement component-based systems to coordinate the functionality of all the components. Components may be reused but the custom logic that binds them into an operational system must still be specified and coded requiring additional consulting and personnel expenses. Therefore, by itself componentized software is not a solution to the above problem.

[0008] One attempted solution to the problem has been described in U.S. Patent No. 6,016,394, which discloses a method and system for database application software creation with minimal programming. This type of system consists of a surrogate environment, which is used to capture the requirements for the target application; a code generator, which synthesizes the code and creates the database for the target application based on the information from the surrogate environment, and the target application itself.

[0009] However, the use of this approach introduces a large overhead and imposes

certain restrictions that are idiosyncratic to each code synthesizer. Most insurance companies prefer a system that does not use a code synthesizer.

[0010] The present invention is therefore directed to the problem of developing a method and apparatus for building software based systems for business solutions, which method and apparatus enable these software-based systems to be developed and fielded rapidly while providing the ability to accept modifications and new requirements in short order, yet reduce the associated costs of development and maintenance while avoiding the use of a code synthesizer.

SUMMARY OF THE INVENTION

[0011] The present invention solves these and other problems by providing an adaptive software application that consists of several types of modules, called Adaptive Units, which are highly parameterized such that they can adapt to varying business requirements by virtue of externally provided parameters. An Adaptive Application is assembled through repeated use of various combinations of different types of Adaptive Units. Large and complex business systems can be rapidly implemented through this approach.

[0012] According to one aspect of the present invention, an Adaptive Unit includes an interface component that can present information to and accept information from the outside world (such as a web page or a system interface), a processing logic component that can manipulate and evaluate information based on parameters received (such as comparisons and decisions as in the case of data dependency decisions), and a data persistence logic component that retrieves, adds, updates, and deletes data targeting one or more Occurrence Databases.

[0013] According to another aspect of the present invention, all three components of an Adaptive Unit are parameter driven. These parameters are not specific to any particular business. One embodiment for providing parameters to these components includes a Definition Database, which is detailed below.

[0014] According to still another aspect of the present invention unit there may be

one or more Occurrence Databases depending on the number of business processes supported by the business application for which the Adaptive Application is being adapted. Typically, there is one Occurrence Database per business process.

[0015] According to yet another aspect of the present invention, the Occurrence Database includes generic (also referred to as meta model driven) tables that are not specific to any particular business.

[0016] The Adaptive Application disclosed herein accelerates business software implementation, reduces the cost of system development and system evolution, improves overall system quality and usability, enables a business to take advantage of previously unapproachable niche product opportunities, and mitigates project risks associated with large systems initiatives.

[0017] While the Adaptive Application invention can be employed for many types of business applications, an exemplary embodiment of an apparatus can be used for creating, managing and administering insurance policies, which is referred to as PolicyWriter. This embodiment can be programmed in Java and HTML code. The system is highly portable and can utilize any JDBC compliant data base management system (e.g., DB2, Oracle, Sybase, MS SQL Server). The embodiment includes an XML oriented database architecture making it readily able to provide enterprise application integration with other systems and communicate externally with other business partners.

[0018] According to one aspect of the present invention, embodiments herein utilize software templates that make it possible to reuse all of the business logic without limiting the ability to customize these templates as needed. This represents a major leap to what is currently available in the marketplace.

[0019] The embodiment is a comprehensive policy administration system that offers unprecedented speed, flexibility, and functionality at an attractive price. It enables a significant reduction in planned costs, time, and resources.

[0020] Other aspects of the present invention will become apparent based upon a reading of the following detailed description in light of the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] FIG 1 depicts a block diagram of an exemplary embodiment of an Adaptive Unit according to one aspect of the present invention.

[0022] FIG 2 depicts a data structure of an exemplary embodiment of a flexible database used in various embodiments for single business processes according to one aspect of the present invention.

[0023] FIG 3 depicts a data structure of an exemplary embodiment of a flexible database used in various embodiments for multiple business processes according to one aspect of the present invention.

[0024] FIG 4 depicts a data structure of another exemplary embodiment of a flexible database used in various embodiments for multiple business processes with relative customization according to one aspect of the present invention.

[0025] FIG 5 depicts a table of the properties by Adaptive Unit according to another aspect of the present invention.

[0026] FIG 6A depicts the process outline for the Add event of the List Adaptive Unit according to another aspect of the present invention.

[0027] FIG 6B depicts the process outline for the Delete event of the List Adaptive Unit according to another aspect of the present invention.

[0028] FIG 7 depicts the process outline for the View/Edit event of the List Adaptive Unit according to another aspect of the present invention.

[0029] FIGs 8A-C depict the process outline for the Load event of the Detail Adaptive Unit according to another aspect of the present invention.

[0030] FIGs 9A-C depict the process outline for the New event of the Detail Adaptive Unit according to another aspect of the present invention.

[0031] FIG 9D depicts the process outline for the Load event of the List Adaptive Unit according to another aspect of the present invention.

[0032] FIGs 10A-B depict the process outline for the Continue event of the Detail Adaptive Unit according to another aspect of the present invention.

[0033] FIGs 11A-B depict an exemplary embodiment of a process for rapidly converting an existing application into a company specific insurance policy management application according to still another aspect of the present invention.

- [0034] FIG 11C depicts the process outline for the Submit event of the Detail Adaptive Unit according to another aspect of the present invention.
- [0035] FIGs 11D-E depict the process outline for the Load event of the Dual List Adaptive Unit according to another aspect of the present invention.
- [0036] FIG 11F depicts the process outline for the View/Edit event of the Dual List Adaptive Unit
- [0037] FIG 11G depicts the process outline for the Add>> event of the Dual List Adaptive Unit according to another aspect of the present invention.
- [0038] FIG 11H depicts the process outline for the <<Remove event of the Dual List Adaptive Unit according to another aspect of the present invention.
- [0039] FIGs 11I-K depict the process outline for the Load event of the Updateable List Adaptive Unit according to another aspect of the present invention.
- [0040] FIG 11L depicts the process outline for the Add Row event of the Updateable List Adaptive Unit according to another aspect of the present invention.
- [0041] FIG 11M depicts the process outline for the Delete Row event of the Updateable List Adaptive Unit according to another aspect of the present invention.
- [0042] FIG 11N depicts the process outline for the Submit event of the Updateable List Adaptive Unit according to another aspect of the present invention.
- [0043] FIG 12 depicts an exemplary embodiment of a header area of a web page employed as a user interface.
- [0044] FIG 13 depicts an exemplary embodiment of a context area of a web page employed as a user interface.
- [0045] FIG 14 depicts an exemplary embodiment of a navigation bar of a web page employed as a user interface.
- [0046] FIG 15 depicts an exemplary embodiment of a signature area of a web page employed as a user interface.
- [0047] FIG 16 depicts an exemplary embodiment of a function bar of a web page employed as a user interface.
- [0048] FIG 17 depicts an exemplary embodiment of an error display area of a web page employed as a user interface.
- [0049] FIG 18 depicts an exemplary embodiment of a main body of a web page

employed as a user interface.

[0050] FIG 19 depicts an exemplary embodiment of a List Adaptive Unit according to another aspect of the present invention.

[0051] FIG 20 depicts an exemplary embodiment of a Detail Adaptive Unit according to another aspect of the present invention.

[0052] FIG 21 depicts an exemplary embodiment of an Updateable List Adaptive Unit according to another aspect of the present invention.

[0053] FIG 22 depicts an exemplary embodiment of a Dual List Adaptive Unit according to another aspect of the present invention.

[0054] FIG 23 depicts an exemplary embodiment of a List-Detail combination Adaptive Unit, which is a combination of a List Adaptive Unit and a Detail Adaptive Unit, according to another aspect of the present invention.

[0055] FIG 24 depicts an exemplary embodiment of a Dual List-Detail combination Adaptive Unit, which is a combination of a Dual List Adaptive Unit and a Detail Adaptive Unit, according to another aspect of the present invention.

[0056] FIG 25 depicts an exemplary embodiment of a Detail-List combination Adaptive Unit, which is a combination of a Detail Adaptive Unit and a List Adaptive Unit, according to another aspect of the present invention.

[0057] FIGs 26-27 depict exemplary embodiments of a Linked Question Source according to another aspect of the present invention.

[0058] FIGs 28-30 depict exemplary embodiments of a Derived Question Source according to another aspect of the present invention.

[0059] FIGs 31-33 depict exemplary embodiments of a Defaulted Question Source according to another aspect of the present invention.

[0060] FIG 34 depicts an exemplary embodiment of a Question Relation of Type A according to another aspect of the present invention.

DETAILED DESCRIPTION

[0061] It is worthy to note that any reference herein to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

[0062] As stated above, the inventions herein are applicable to many types of business applications, including but not limited to insurance policy management, financial services and product design, management and implementation, legal services, industrial processes, architectural design, etc. to name only a few. Other applications will become apparent based on the description herein.

[0063] To facilitate an understanding of the various inventions herein, an embodiment relating to management of insurance policies will now be described. Many of the steps in creating the necessary application require interaction with the customer to determine the requirements of the particular application. These requirements are embodied in the navigational design of the web pages and the layout of the various web pages. Once the navigational design and layouts are complete, the inventions herein enable rapid development of the necessary web screens. This will become apparent in the following exemplary embodiment.

Overall Process for Creating Insurance Policy Management Application

[0064] The present invention includes *inter alia* an adaptive application that can be adapted to a wide variety of business applications, including insurance and financial services, to name only a few. The adaptive application includes the core modules needed to perform the main functionality of any business application.

[0065] Turning to FIGs 11-12, a process for adapting the software application to manage policies for an insurance company is shown therein. In this exemplary embodiment, the resulting application must create, manage and administer a variety of insurance products, some of which may be developed after the creation of this software application.

[0066] The first step 101 of adapting the adaptive application for an insurance

company to manage its policies consists of identifying the screens needed for entering and modifying the information to describe the insurance products (e.g., what coverages, limits, deductibles are offered) to be managed by the application. Typically, this step is part of an interview process with the various employees and managers of the insurance company. In conventional software developments, this step is often employed later in the process, but the inventors have determined that by identifying these screens first, the application can be built more rapidly than heretofore possible.

[0067] Once the screens are identified, the web pages can be created. According to one aspect of the present invention, the inventors have identified that any web page needed in these business applications can be created from various combinations of Adaptive Units, which will be described in more detail below. By reusing the Adaptive Units in necessary and/or desired combinations, any web page or screen can be created in a fraction of the time heretofore possible. This significantly speeds up the process of developing these types of large software applications. Moreover, while the Adaptive Units are described herein with respect to web pages, Adaptive Units can be configured to interact with other interfaces.

[0068] The Adaptive Units set forth herein provide three levels of functionality – interfacing with a user, performing data manipulation and performing data persistence functions with the Occurrence Database. These functions will be described more completely later. There are several different types of Adaptive Units, which are discussed below. By combining each type of Adaptive Unit with other types, as necessary, any interface screen or device can be created, thereby removing the requirement for developing these separately.

[0069] For each screen identified in Step 1 (element 101), next the Adaptive Units are created (element 102), which are copies of Adaptive Unit Templates that apply to that screen. In this case, the user identifies the necessary functionality and uses the various Adaptive Unit types to create the necessary functionality of the interface.

[0070] A View Name is assigned to each Adaptive Unit by inserting it into the copied software (element 103). Each Adaptive Unit uses this View Name to obtain all of its parameters. Some Adaptive Units may require two View Names. The View Name

links externally supplied parameters to the particular Adaptive Unit, thereby enabling the Adaptive Unit to be adapted to a specific function desired by the user.

[0071] In the Definition Database, the necessary information associated with each View Name that has been assigned is created (element 104). This information includes VIEW, ENTITY, ENTITY ANCESTRY, QUESTIONS, QUESTION RELATIONS, ANSWER OPTIONS, VIEW ITEMS, VIEW ITEM DEPENDENCY LINES, and VIEW ITEM DEPENDENCY VALUES (see FIG 2). All of these information elements are not specific to any given business process, and therefore can be filled with information necessary to any given business process. Thus, the Adaptive Units can receive parameters that are unique to each business process to which the application is being adapted through these information fields from, for example the Definition Database. This information could be provided by another method, such as via an application programming interface to name only one possibility.

[0072] Next, all the Adaptive Units are assembled into a "Product Definition" application (element 105). In other words, this Product Definition application is the first application for which the Adaptive Application was adapted. This Product Definition application creates the web pages or screens and functionality that were identified in step 101.

[0073] The Product Definition Application is then used to enter the information to describe each insurance product (106).

[0074] Now the process moves to FIG 11 (element 107), which depicts the process for adapting the Adaptive Application into an application for creating insurance policies 110.

[0075] At this point it becomes necessary to identify the screens needed for entering and modifying the information to describe policies (e.g., insured's name, address, selected coverages, limits and deductibles) (element 111).

[0076] For each screen identified in the immediately preceding paragraph, the Adaptive Units are created (element 112), which Adaptive units are copies of the Adaptive Unit Templates that apply to that screen.

[0077] A View Name is then assigned to each newly created Adaptive Unit, by

inserting it into the copied software (element 113). The Adaptive Unit will use this View Name to obtain all of its parameters. As before, some Adaptive Units may require two View Names.

[0078] In the Definition Database, the necessary information associated with each View Name that has been assigned is created (element 114). This information consists of two types. The cross-product information includes VIEW, ENTITY, ENTITY ANCESTRY, QUESTIONS, QUESTION RELATIONS, ANSWER OPTIONS, VIEW ITEMS, VIEW ITEM DEPENDENCY LINES, and VIEW ITEM DEPENDENCY VALUES. All of these information elements are not specific to any given business process, and therefore can be filled with information necessary to a given business process. Thus, the Adaptive Units for this business process can receive parameters that are unique to each business process to which the application is being adapted through these information fields from, for example the Definition Database. This information could be provided by another method, such as via an application programming interface to name only one possibility. The product specific information includes CONTRACT QUESTION BY PRODUCT, CONTRACT QUESTION BY PRODUCT OCCURRENCE, and CONTRACT ANSWER OPTION BY PRODUCT OCCURRENCE. These information elements make it possible to vary the definition parameters for one business process (in this case, Contract Sales) with respect to the elements of another business process (in this case, Product Development). As a result, in this embodiment, the parameters that govern the behavior of the Adaptive Application for selling contracts can be varied by product or a subcomponent of a product such as Coverage Type.

[0079] Next, all the Adaptive Units are assembled into the "Policy Writing" application (element 115).

[0080] Then, the Policy Writing Application is used to write policies (element 116). Thus, this process 110 ends (element 117).

System Components

[0081] The embodiments of the present invention include several elements: Adaptive Units, Definition parameters, and one or more Occurrence Databases.

[0082] According to one aspect of the present invention, the business is reduced to a linear flow of business processes. An occurrence database is assigned to each process of the flow. Links between subsequent processes in the linear flow of the business are established to provide details in subsequent processes that are determined by earlier processes of the overall business.

[0083] For example, in the insurance policy management business, a process for creating the insurance products is identified and an occurrence database is defined and associated with this process. A second process is identified as the insurance policy writing process that can only occur based on insurance products defined in the earlier process of product definition. This business process identification can then continue through the entire operation of the business.

Database

[0084] FIG 2 shows an exemplary embodiment of a data structure 20 for the database used in the various embodiments herein. The database is composed of at least two parts – a definition database 21 and one or more occurrence databases 22. A data dictionary is included herein as Appendix A. The terms used below are defined more completely therein. The relationship and interlinks between these databases are shown in FIG 1. The Occurrence Database 22 consists of generic (also referred to as meta model driven) tables that are not specific to any business process.

[0085] The notation linking the various blocks is in standard form, which indicates the relationship between various blocks. For example, each entity may have zero (indicated by the zero in the line linking the two blocks), one (indicated by the one written across the line linking the two blocks) or more contract occurrences (indicated by the “crows feet” symbol at the end of the link between the two blocks). Moreover, each contract occurrence has one and must have one entity. Thus, this notation depicts the relationship between the two blocks. Each block represents a table in the database. Each table may have one or more entries. The links between boxes are links between the tables in the database.

[0086] The Definition Database 21 includes an Entity table 21a, to which there may

be linked tables for Entity Ancestry 21b, View 21c, and Question 21d. In turn, the View table 21c may be linked to multiple tables of View Items 21f, which may be linked to one or more tables of View Item Dependency Lines 21e and which in turn may be linked to one or more tables of View Item Dependency Values 21i. There may be links between the table of Questions 21d and the View Item table 21f, as well as to the Question Relation table 21g and the Answer Option table 21h.

[0087] Thus, in FIG 2, the Occurrence Database 22 has a Master Business Process (BP) Occurrence Table 22a, to which may be linked multiple tables of Business Process Occurrences 22b, which in turn may have multiple tables of Business Process Answers 22c and Business Process Ancestry 22d.

[0088] One or more Business Process Occurrence tables 22b may be linked to the Entity table 21a. The Business Process Answer tables 22c may be linked to the Question table 21d.

[0089] FIG 3 depicts the data structure 30 for the database in a situation in which there is more than one business process. In this case, there is at least one Occurrence Database for each business process. This data structure 30 includes a Definition Database 31 and two Occurrence Databases 32-1, 32-2, one for Business Process 1 and another for Business Process 2. The only difference between this data structure 30 and that shown in FIG 2, other than the second Occurrence Database 32-2, is additional links 33, 34 between the Definition Database and the two Occurrence Databases 32-1, 32-2.

[0090] FIG 4 depicts the Adaptive Application Data Structure 40 with Multiple Business Processes and Relative Customization. The data structure 40 in this version has been modified to customize the Definition Database 41 so that an upstream business process can affect answers to downstream business processes. Thus, there are three additional tables provided in the Definition Database 41, which include Business Process 2 Question by Business Process 1 42a, Business Process 2 Question by Business Process 1 Occurrence 42c, and Business Process 2 Answer Option by Business Process 1 Occurrence 42b. In this case, the Business Process 1 Occurrence Database is for an upstream Business Process, which is upstream in the business relative to Business Process 2. Thus, the Business Process 1 to which a particular Business Process 2 is applicable can

alter questions (i.e., fields) and answers in the Definition Database 41. Additional links 43a-h are necessary to supply the required functionality.

Definition Database

[0091] Throughout the description there will be reference made to Product, which is a first business process (BP1) in the insurance business, and Contract, which is a second business process (BP2) in the insurance business relative to Product. The names are used in certain instances to provide more meaning to the reader than the generic names, however, these tables are not specific to any business and can be employed to adapt the Adaptive Application to any desired business.

[0092] The Entity Table specifies details of each entity defined in the logical model. For example, PRODUCT, PRODUCT COVERAGE, POLICY, POLICY COVERAGE are defined therein for a business process of PRUDUCT in the insurance business.

[0093] The Entity Ancestry Table specifies the parent-child relationship between two entities in the conceptual model. For example, there is a parent-child relationship between POLICY & POLICY LOCATION in the insurance business.

[0094] The Question Table specifies each "Question" associated with an Entity. The View Table normally corresponds to one Adaptive Unit. Sometimes more than one Adaptive Unit utilizes the same view. An Interface (a web page or an external interface) consists of one or more Views. Each View consists of a selection of elements from one "Entity."

[0095] This Contract (BP2) Question of a Product (BP1) Table specifies the properties for a Contract (BP2) Question of a Product (BP1).

[0096] The Question Relation Table specifies the relationship between two Questions.

[0097] The View Item Table defines each item in a View. The View Item can be an Image, a Literal (i.e., text) or a Question with a number of properties that specify its label and control the capture and display of its answer. The View Item specifies the presentation characteristics for each displayed element.

[0098] The View Item Dependency Line Table defines the dependencies or the

condition under which the View Item (e.g., display element) is presented. This provides a way by which a display element can be presented only under certain conditions.

Dependency Line along with Dependency Value enables specification of any complex expression. The Dependency Expression is a complex Boolean expression with the left side operand is a Question and the right side operand can be a literal or a Question. The comparison operator can be one of <, >, <=, >=, <>, =. The operations can be grouped by appropriate parentheses. For example a valid expression is ((Coverage=C1 or C2 or C3) AND (Vehicle Type = T1 or T2 or T3)).

[0099] This Dependency Value Table defines the set of values that need to be satisfied with Dependency Line Primary Q. Each instance specifies one value of the OR expression (in the example C1, C2 and C3). Each value may be a constant or another Question (specified here as the Secondary Question).

[0100] The Master Product (BP1) Occurrence Table contains one record for every instance of a Product (BP1).

[0101] The Product (BP1) Occurrence Table contains one record for each occurrence of each entity in the Product (BP1) hierarchy.

[0102] The Product (BP1) Ancestry Table identifies the parent-child relationships between Product (BP1) Occurrences.

[0103] The Product (BP1) Answer Table contains the values for the individual attributes of a Product (BP1) Occurrence.

[0104] The Long Product (BP1) Answer Table contains the Product (BP1) Answer value if the Product (BP1) Answer's size exceeds 20 characters.

[0105] The Contract (BP2) Question by Product (BP2) Occurrence Table specifies the overriding values for presentation characteristics of a Question based on Product (BP1) Occurrence.

[0106] The Product (BP1) Answer Option Table specifies the possible values for the Answer of a Question relative to a particular Product (BP1) Occurrence Question. If the Product (BP1) Occurrence Question Answer Option Category is 1 (specific list), each possible value is specified in this table, one record per value. If it is 2 (ranges of values), then each range must be specified in one record using From, To and Increment.

[0107] The Answer Option Table specifies the possible values for all Questions

including variations by Product. If the Question Answer Option Category is 1 (specific list), each possible value is specified in this table, one record per value. If it is 2 (ranges of values), then each range must be specified in one record using From, To and Increment.

[0108] The Master Contract (BP2) Occurrence Table contains one record for every instance of a Contract (BP2). This record ties together all the details for the Contract (BP2).

[0109] The Contract (BP2) Occurrence Table contains one record for each occurrence of each entity in the Contract (BP2) hierarchy.

[0110] The Contract (BP2) Ancestry Table identifies the parent-child relationships between Contract (BP2) Occurrences.

[0111] The Contract (BP2) Answer Table contains the values for the individual attributes of a Contract (BP2) Occurrence.

[0112] The Long Contract (BP2) Answer Table contains the Contract (BP2) Answer value if the Contract (BP2) Answer's size exceeds 20 characters.

User Interface

[0113] Turning to FIG 12, every user interface that is manifest as a web page 120 typically includes the following elements – a header area 121 (see FIG 12), a context area 131 (see FIG 13), a navigation bar 141 (see FIG 14), a signature area 151 (see FIG 15), a function bar 161 (see FIG 16), an error display area 171 (see FIG 17), and a main body 181 (see FIG 18). The focus herein relates to the main body 181.

[0114] According to one aspect of the present invention, the main body 181 of a web page 120 employed as a user interface can be composed of several reusable elements in various combinations. We term these “Adaptive Units.” An exemplary embodiment 16 of an Adaptive Unit is shown in FIG 1A. These Adaptive Units include three components – an interface component 12, a data persistence component 13 and a data processing component 14.

[0115] The interface component 12 presents information to and accepts information from the outside world (such as a web page or a system interface).

[0116] The processing logic component 13 can manipulate and evaluate information

based on received parameters. This component can perform complex manipulations such as comparisons and decisions as in the case of data dependency decisions.

[0117] The data persistence logic component 14 retrieves, adds, updates, and deletes data targeting one or more Occurrence Databases 15a-15n. This component manages the data in the one or more Occurrence Databases.

[0118] All three components of an Adaptive Unit are parameter driven. In other words, each of these components receives parameters from somewhere else that drive the activities of the component. These parameters are embedded throughout these components so that the components can be adapted to any application. As a result, these components are highly flexible forming the core of the Adaptive Application.

[0119] One embodiment for providing parameters to these components includes a Definition Database 11. The Definition Database can be populated with data specific to the given application for which the Adaptive Application is being adapted to completely control the three components of each Adaptive Unit. By using these Adaptive Units as building blocks the Adaptive Application can be developed in short order.

Adaptive Units

[0120] There are multiple types of Adaptive Units (AU's) that can be combined in various manners to create any combination of web page necessary for interacting with a user. Some embodiments of these Adaptive Units include -- a List AU (an example of which is shown in FIG 19), a Detail AU (an example of which is shown in FIG 20), an Updateable List AU (see FIG 21 for an example) and a Dual List AU (see FIG 22 for an example), which only becomes necessary when there are multiple occurrence databases.

[0121] Referring to FIG 19, shown therein is a List Adaptive Unit 190, which includes a list 195 and several events (Add, Delete, Detail, Reset) that can be initiated by clicking on the buttons 191-194, respectively.

[0122] Referring to FIG 20, shown therein is a Detail Adaptive Unit that includes a detail area 201 and a function button 202 (submit).

[0123] Referring to FIG 21, shown therein is an Updateable List Adaptive Unit that includes a list 214 and several function buttons (Add Row, Delete Row, Submit) 211-213, respectively.

[0124] Referring to FIG 22, shown therein is a Dual List Adaptive Unit that includes a left side list 221 and a right side list 222.

[0125] Referring to FIG 23, shown therein is a List Detail Unit that includes a list 232 and a detail area 231 for the selected item on the list.

[0126] Referring to FIG 24, shown therein is a Dual-List Detail Adaptive Unit that includes a left side list 241 and a right side list 243 and a detail area 242 for the selected item.

[0127] Referring to FIG 25, shown therein is a Detail – List Adaptive Unit that includes a detail area 251 and a list 252 for the selected item on the detail area 251.

[0128] Sometimes the main body of the web page is composed of a combination of two or more Adaptive Units. The following are some examples of frequently encountered combinations -- List-Detail AU (see FIG 23 for an example), Dual-List Detail AU (see FIG 24 for an example), Detail-List AU (see FIG 25 for an example). Other combinations are also possible depending upon the precise requirements of a given application. Essentially, any desired web page can be created from a combination of these four Adaptive Units, which makes it rather simple to create any desired web page rapidly.

List Adaptive Unit

[0129] In the List Adaptive Unit, there are several possible events that can be initiated – Load (see FIG 9D), View/Edit (see FIG 7), Add (see FIG 6A) and Delete (see FIG 6B). Each event is activated when a user clicks on the link associated with the given event. The List Adaptive unit has the following properties – ViewName, DetailTarget, ProductId and Occurrence ID. These properties are the information elements via which the List Adaptive Unit receives and transfers parameters.

[0130] Turning to FIG 6A, shown therein is the process 60a that is initiated upon clicking on the Add link in the List Adaptive Unit. This process 60a in turn activates the “New” event of the Detail Adaptive Unit (see FIGs 9A-C) using DetailTarget, ProductId, and OccurrenceId of the parent occurrence for the list (step 61a). The remaining steps are explained in the New event of the Detail Adaptive Unit below.

[0131] Turning to FIG 6B, shown therein is the process 60b that is initiated upon

clicking on the Delete link in the List Adaptive Unit. First, the Contract Occurrence with the OccurrenceId is deleted (step 61b). Then, the deleted occurrence is removed from the display (step 62b).

[0132] Turning to FIG 7, shown therein is the process 70 that is initiated upon clicking on the View/Edit link in the List Adaptive Unit. This process 70 in turn activates the “Load” event of the Detail Adaptive Unit (see FIGs 8A-C) using DetailTarget, ProductId, and OccurrenceId of a selected item (step 71). The remaining steps are explained in the Load event of the Detail Adaptive Unit below.

[0133] Turning to FIG 9D, the process 90d initiated by clicking on the Load link in the List Adaptive Unit is shown therein. Once initiated, the process 90d retrieves the View attributes using the ViewName (step 91d). Next, the attributes of each View Item are retrieved for the ViewName and ProductId (step 92d). For each View Item, the Question attributes are retrieved (step 93d). Using the OccurrenceId as a “parent-id”, the “children” Contract Occurrences are retrieved from the Contract Ancestry (step 94d). Next, the Contract Answers are retrieved for the Questions related to the View Items from each occurrence (step 95d). Finally, the answers are displayed for each Contract Occurrence in a separate line (step 96d) and the process ends.

Detail Adaptive Unit

[0134] In this Adaptive Unit, there are several possible events that can be initiated – Load, New, Continue and Submit. The Detail Adaptive Unit has the following properties – ViewName, Occurrence ID, ProductId and SegmentNo.

[0135] Referring to FIGs 8A-C, shown therein is the process 80 that is initiated upon clicking on the Load link of the Detail Adaptive Unit. The SegmentNo is initialized to one (1) 81a. The View Attributes are retrieved using the ViewName 82a. The attributes for each View Item for the View Name and ProductId are then retrieved 83a. Next, for each View Item, the View Item Dependency Lines and the View Item Dependency Values are retrieved 84a. For each View Item related to a Question, steps 86a-89a and 81b-88b are performed, after which the process moves to step 89b. In step 86a, the question attributes are retrieved. In step 87a, the SegmentNo is compared to the Segment of the View Item, and if equal, the process moves to step 88a (FIG 8B). If the

SegmentNo is not equal to the Segment of the View Item the process moves to step 88b of FIG 8B (step 89a).

[0136] Turning to FIG 8B, the process continues in step 81b from FIG 8A. Next, the ProductId is evaluated and if the ProductId is not zero, the process moves to step 83b, in which the Question attributes are overridden using the Contract Question by Product and the process moves to step 86b. If the ProductId is zero, the process moves to step 84b, in which the determination is made as to whether the question has answer options, and if so, the process moves to step 85b, in which the attributes of each answer option are retrieved, and then the process moves to step 86b. If the Question does not have answer options, the process moves to step 86b directly, thereby skipping step 85b. Next, the determination is made as to whether the question has question relations 86b. If the question has question relations, then the process moves to step 87b, which takes us to FIG 8C, step 81c. If the question does not have question relations, then the process moves to step 89b, in which after eliminating those View Items that are inapplicable because of question dependency conditions, the remaining View Items are displayed with Segment equal to SegmentNo along with answers that are not hyperlinked, and the process ends.

[0137] Turning to FIG 8C, this subprocess is initiated by an affirmative response to the determination of step 86b. First, the attributes of each Question Relation are retrieved 82c. For “default” and “derive” relations, the Contract Answer is retrieved from higher-level contract entities using OccurrenceId and Contract Ancestry to determine value of the answer 83c. For a “link” question relation, the set of possible selections is retrieved using the ProductId 84c. For a “dynamic” question relation, attributes from “Contract Question By Product Occurrence” and from “Contract Answer Option by Product Occurrence” are retrieved 85c, and the process returns to FIG 8B, step 87b (step 86c).

[0138] Referring to FIGs 9A-C, shown therein is the process 90 that is initiated upon clicking on the New link of the Detail Adaptive Unit. The SegmentNo is initialized to one (1) 91a. The View Attributes are retrieved using the ViewName 92a. The attributes for each View Item for the View Name and ProductId are then retrieved 93a. Next, for each View Item, the View Item Dependency Lines and the View Item Dependency Values are retrieved 94a. For each View Item related to a Question, steps 96a-99a and 91b-98b are performed, after which the process moves to step 99b. In step 96a, the question attributes

are retrieved. In step 97a, the SegmentNo is compared to the Segment of the View Item, and if equal, the process moves to step 98a (FIG 9B). If the SegmentNo is not equal to the Segment of the View Item the process moves to step 98b of FIG 9B (step 99a).

[0139] Turning to FIG 9B, the process continues in step 91b from FIG 9A. Next, the ProductId is evaluated and if the ProductId is not zero, the process moves to step 93b, in which the Question attributes are overridden using the Contract Question by Product and the process moves to step 96b. If the ProductId is zero, the process moves to step 94b, in which the determination is made as to whether the question has answer options, and if so, the process moves to step 95b, in which the attributes of each answer option are retrieved, and then the process moves to step 96b. If the Question does not have answer options, the process moves to step 96b directly, thereby skipping step 95b. Next, the determination is made as to whether the question has question relations 96b. If the question has question relations, then the process moves to step 97b, which takes us to FIG 9C, step 91c. If the question does not have question relations, then the process moves to step 99b, in which after eliminating those View Items that are inapplicable because of question dependency conditions, the remaining View Items are displayed with Segment equal to SegmentNo along with answers that are not hyperlinked, and the process ends.

[0140] Turning to FIG 9C, this subprocess is initiated by an affirmative response to the determination of step 96b. First, the attributes of each Question Relation are retrieved 92c. For “default” and “derive” relations, the Contract Answer is retrieved from higher-level contract entities using OccurrenceId and Contract Ancestry to determine value of the answer 93c. For a “link” question relation, the set of possible selections is retrieved using the ProductId 94c. For a “dynamic” question relation, attributes from “Contract Question By Product Occurrence” and from “Contract Answer Option by Product Occurrence” are retrieved 95c, and the process returns to FIG 9B, step 97b (step 96c).

[0141] Referring to FIGs 10A-B, shown therein is the process 100a that is initiated upon clicking on the Continue link of the Detail Adaptive Unit. First, the SegmentNo is increased by one (1) 101a. Next, for each View Item with Segment=SegmentNo that is related to a Question, steps 103a-109a are performed, after which the process moves to step 110a (step 102a). In step 103a, the ProductId is evaluated and if the ProductId is not zero, the process moves to step 104a, in which the Question attributes are overridden

using the Contract Question by Product and the process moves to step 107a. If the ProductId is zero, the process moves to step 105a, in which the determination is made as to whether the question has answer options, and if so, the process moves to step 106a, in which the attributes of each answer option are retrieved, and then the process moves to step 107a. If the Question does not have answer options, the process moves to step 107a directly, thereby skipping step 106a. Next in step 108a, the determination is made as to whether the question has question relations. If the question has question relations, then the process moves to step 109a, which takes us to FIG 10B, step 101b. If the question does not have question relations, then the process moves to step 110a, in which the View Items with Segment less than SegmentNo are displayed along with hyperlinked answers. In step 111a, after eliminating those View Items that are inapplicable because of question dependency conditions, the remaining View Items are displayed with Segment equal to SegmentNo along with answers that are not hyperlinked, and the process ends.

[0142] Turning to FIG 10B, this subprocess is initiated by an affirmative response to the determination of step 108a. First, the attributes of each Question Relation are retrieved 102b. For “default” and “derive” relations, the Contract Answer is retrieved from higher-level contract entities using OccurrenceId and Contract Ancestry to determine value of the answer 103b. For a “link” question relation, the set of possible selections is retrieved using the ProductId 104b. For a “dynamic” question relation, attributes from “Contract Question By Product Occurrence” and from “Contract Answer Option by Product Occurrence” are retrieved 105b, and the process returns to FIG 10A, step 109a (step 106b).

[0143] Referring to FIG 11C, shown therein is the process 112c that is initiated upon clicking on the Submit link of the Detail Adaptive Unit. First, in step 113c, if a new Contract Occurrence has been introduced, the new Contract Occurrence is added to the database. Then in step 114c, each new Contract Answer is added to the database. Each Contract Answer that has been modified in the database is then updated (step 115c). Next, questions that have been deleted or made inapplicable (by dependency conditions) from the database are deleted in step 116c.

Dual List Adaptive Unit

[0144] In this Adaptive Unit, there are four possible events that can be initiated – Load, View/Edit, Add >> and <<Remove. This Adaptive Unit has the following properties – LeftViewName, RightViewName, DetailTarget, LinkedEntity, LinkedQuestion, Occurrence ID and ProductId.

[0145] Referring to FIGs 11D-E, shown therein is the process that is initiated upon clicking on the Load link of the Dual List Adaptive Unit. FIG 11D shows the portion 112d of the process that is applicable to the list on the right side of the selected items, whereas FIG 11E shows the portion 112e of the process that is applicable to the list on the left side of the selected items.

[0146] Turning to FIG 11D, in step 113d, the View attributes are retrieved using the RightViewName. Next in step 114d, the attributes of each View Item are retrieved for the RightViewName and ProductId. Next in step 115d, for each View Item, the Question attributes are retrieved. In step 116d, using OccurrenceId as a “parent-id”, the “children” Contract Occurrences are retrieved from the Contract Ancestry. Next in step 117d, Contract Answers are obtained for the Questions related to the View Items from each occurrence. Finally in step 118d, the answers for each Contract Occurrence are displayed in a separate line.

[0147] Turning to FIG 11E (for the list on the left side of the selectable items), in step 113e, the View attributes are retrieved using the LeftViewName. Next in step 114e, the attributes of each View Item are retrieved for the LeftViewName and ProductId. Next in step 115e, for each View Item, the Question attributes are retrieved. In step 116e, all Product Occurrences are found that contain a Product Answer for the question identified by TargetEntity and TargetQuestion. Next in step 117e, Contract Answers are obtained for the Questions related to the View Items from each occurrence. Finally in step 118e, the answers for each Product Occurrence are displayed in a separate line excluding the occurrences that are already selected and therefore in the “selected items list” (i.e., matching values for the LinkedQuestion and TargetQuestion).

[0148] Referring to FIG 11F, shown therein is the process that is initiated upon

clicking on the View/Edit link of the Dual List Adaptive Unit. Clicking the View/Edit link initiates process 110f, which in step 111f activates the “Load” event of the Detail Adaptive Unit using DetailTarget, ProductId, and OccurrenceId (of selected item).

[0149] Referring to FIG 11G, shown therein is the process that is initiated upon clicking on the “Add>>” link of the Dual List Adaptive Unit. Clicking the “Add>>” link initiates process 110g, which in step 111g activates the “New” event of the Detail Adaptive Unit using DetailTarget, ProductId, and OccurrenceId (of the parent occurrence for the list).

[0150] Referring to FIG 11H, shown therein is the process that is initiated upon clicking on the “Remove>>” link of the Dual List Adaptive Unit. Clicking the “Remove>>” link initiates process 110h, which in step 111h deletes the Contract Occurrence with OccurrenceId. Next in step 112h, the deleted occurrence is removed from display on the right list. Finally, in step 113h, the corresponding occurrence is displayed on the left list.

Updateable List Adaptive Unit

[0151] In this Adaptive Unit, there are several possible events that can be initiated – Load, Add Row, Delete Row and Submit. This Adaptive Unit has the following properties – ListViewName, DetailViewName, Occurrence ID and ProductId.

[0152] Referring to FIGs 11I-K, shown therein is the process 110i that is initiated upon clicking on the Load link of the Updateable List Adaptive Unit. In step 111i, the View Attributes are retrieved using the ListViewName. The attributes for each View Item for the ListView Name and ProductId are then retrieved 112i. Next, for each View Item, the attributes are retrieved for the DetailViewName and ProductId 113i. For each View Item from the DetailViewName, steps 115i-116i, and 111j-117j are performed, after which the process moves to step 118j. In step 115i, the question attributes are retrieved. In step 116i, the process moves to FIG 11J, element 111j.

[0153] Turning to FIG 11J, the process continues in step 111j from FIG 11I, step 116i. Next, the ProductId is evaluated and if the ProductId is not zero, the process moves to step 113j, in which the Question attributes are overridden using the Contract Question by Product and the process moves to step 116j. If the ProductId is zero, the process

moves to step 114j, in which the determination is made as to whether the question has answer options, and if so, the process moves to step 115j, in which the attributes of each answer option are retrieved, and then the process moves to step 116j. If the Question does not have answer options, the process moves to step 116j directly, thereby skipping step 115j. Next, the determination is made as to whether the question has question relations 116j. If the question has question relations, then the process moves to step 117j, which takes us to FIG 11K, step 111k. If the question does not have question relations, then the process moves directly to step 118j, in which the “children” Contract Occurrences from the Contract Ancestry are retrieved using OccurrenceId as a “parent-id.” In step 119j, the Contract Answers for the Questions related to the DetailViewName View Items from each occurrence are obtained. Finally in step 120j, the answers for each Contract Occurrence are displayed in a separate line, and the process ends.

[0154] Turning to FIG 11K, this subprocess is initiated by an affirmative response to the determination of step 116j. First, the attributes of each Question Relation are retrieved 112k. For “default” and “derive” relations, the Contract Answer is retrieved from higher-level contract entities using OccurrenceId and Contract Ancestry to determine value of the answer 113k. For a “link” question relation, the set of possible selections is retrieved using the ProductId 114k. For a “dynamic” question relation, attributes from “Contract Question By Product Occurrence” and from “Contract Answer Option by Product Occurrence” are retrieved 115k, and the process returns to FIG 11J, step 117j (step 116k).

[0155] Referring to FIG 11L, shown therein is the process that is initiated upon clicking on the Add Row link of the Updateable List Adaptive Unit. Clicking the Add row link initiates process 110l, which in step 111l adds a blank new row to the updateable list filling in default values, if any (no database change), and the process ends.

[0156] Referring to FIG 11M, shown therein is the process that is initiated upon clicking on the Delete Row link of the Updateable List Adaptive Unit. Clicking the Delete row link initiates process 110m, which in step 111m deletes the indicated row from the updateable list (no database change), and the process ends.

[0157] Referring to FIG 11N, shown therein is the process that is initiated upon

clicking on the Submit link of the Updateable List Adaptive Unit. Clicking the Submit link initiates process 110n, which in step 111n adds each new Contract Occurrence, if any, to the database. Next in step 112n, each new Contract Answer, if any, is added to the database. Then in step 113n, each Contract Answer that has been modified in the database is updated. Finally in step 114n, questions that have been deleted or made inapplicable (by dependency conditions) are deleted from the database, and the process ends.

QUESTION RELATIONS

[0158] One or more of the embodiments of the present invention employ a technique known as Question Relations to provide an efficient, declarative, and non-redundant way for the Adaptive Application (e.g., PolicyWriter) to associate the answer of a Question with that (i.e., an answer) of another Question. The related question can be from the same entity or a different entity.

[0159] Depending on its type, a Question Relation may point to one or more Questions that are associated with a given question: Target-Q or Ref-Q.

[0160] The following table lists the Question Relation types.

| Code | Name | Parameters |
|------|----------|-----------------|
| A | Variable | Target-Q |
| L | Link | Target-Q |
| S | Search | Target-Q |
| V | Value | Target-Q, Ref-Q |

The usage of all these Question Relations types, except for type A, (see variable properties below) is governed by QUESTION SOURCE, a Question property, which enables the answer of a Question to be associated with the answers to other Questions (from the same or different Entities).

[0161] The following QUESTION SOURCE values are of interest: Linked (L), Derived (D), and Defaulted (F).

[0162] Referring to FIG 26, A LINKED Question 262 duplicates a Question 261 of

an “upstream” Entity, thus creating an association between two Entities. The answer of a Linked Question is always stored. Question relations that may be needed to facilitate a linked question are:

L – points to a question in the other database to link to

S – points to a question in the current database that provides the value of the particular parent occurrence under which the answer to the Question to link to is to be found in the other database

V – points to a default answer for the linked question (in the current database)

[0163] Turning to FIG 27, the Linked Question is the Coverage Code 275. The Product Coverage Code is the question being linked to (element 272); Product Code 271 is used to determine the applicable Product Coverages. Two question relations are needed for this example. One of type L to point to the Product Coverage Code 275, and one of type S to use the Policy Product Code 273 as the parent ID for retrieving Product Coverages 272.

[0164] Turning to FIG 28, A DERIVED Question acquires its answer by applying a formula to the answer of an Ancestor’s or Extended Ancestor’s Question which is called the Target Question. The answer of a Derived Question may be stored depending on the “Storage Code” property.

[0165] The Target Question is element 282, or 283. If the Target Question belongs to an Extended Ancestor, it must be specified via a Reference Question which is itself a link (element 284). The derived question is element 281, or 285. Only one Question Relation of type V needed in either case.

[0166] Turning to FIG 29, the Derived Question is Location Limit (element 294). Target Question is the Policy Limit (element 293). One Question Relation of type V is needed.

[0167] Turning to FIG 30, the Derived Question is Coverage Premium Ind (element 305). The Target question is Product Coverage Premium Ind (element 301). The Coverage Code is the link that is used to make the connection (Coverage Ccode in element 305). One Question relation row of type V is needed (Target-Q = Product Coverage.Premium Ind; Ref-Q = Coverage.Coverage Code).

[0168] Turning to FIG 31, A DEFAULTED Question acquires its initial answer by

applying a formula to the answer of an Ancestor's or Extended Ancestor's Question which is called the Target Question. The answer of a Defaulted Question may be stored depending on the "Storage Code" property.

[0169] The Target Question is element 311 or 313. If the Target Question belongs to an Extended Ancestor, it must be specified via a Reference Question which is itself a link (element 314). The defaulted question is element 312 or 315. Only one Question Relation of type V is needed in either case.

[0170] Turning to FIG 32, the Defaulted Question is the Coverage Deductible (element 325). The Target Question is the Policy Deductible (element 323). One Question Relation row of type V is needed (Target-Q = Policy.Deductible).

[0171] Turning to FIG 33, the Defaulted Question is the Coverage Inclusion Ind (element 335). The Target Question is the Product Coverage Inclusion Ind (element 332). The Coverage Code is the link that is used to make the connection (Coverage Code in element 335). One Question Relation of type V is needed (where Target-Q = Product Coverage.Inclusion Ind and Ref-Q = Coverage.Coverage Code).

Variable Properties

[0172] Answer options and several other properties of Contract Questions are referred to as Variable Properties. Answer options of a Contract Question are set at the level of a designated Product Entity. For example, the answer options for the "Coverage limit" question may differ by the "Product Coverage Type". The remaining Variable Properties are set elsewhere but may be overridden at the level of the designated Product Entity. If a Contract Question requires Variable Properties, it must have a Question Relation of type A, which specifies another Contract Question (Target-Q) that provides the link to a particular occurrence of the Product Entity where the Variable Properties reside.

[0173] Turning to FIG 34, the Limit (element 345) is the Contract Question whose variable property level is being designated. In this case, the Question Relation has Coverage Code (Coverage code in element 345) as the Target-Q. Product Coverage (element 342) is the Product Entity that the Target-Q is linked to.

Summary

[0174] Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the invention are covered by the above teachings and are within the purview of the appended claims without departing from the spirit and intended scope of the invention. For example, specific software languages and database types are mentioned, however, the scope of the present invention is not limited to these examples. Furthermore, these examples should not be interpreted to limit the modifications and variations of the invention covered by the claims but are merely illustrative of possible variations.

Appendix A

| Entity | | | |
|--|-----------------------------|---|------------------------|
| This table specifies details of each entity defined in the logical model. For e.g. PRODUCT, PRODUCT COVERAGE, POLICY, POLICY COVERAGE. | | | |
| Field Name | Attribute Definition | Description | Possible Values |
| Id | int | Uniquely Identifies a entity. | |
| Type | T 1 | Identifies the business process (Product Development, Contract Sale) this entity belongs to | P-Product, C-Contract |
| Name | T 50 | Long Descriptive Name of the entity | |
| Standard Reference Code | T 20 | Standard Tag that is associated with the Entity. An example of this is the ACCORD XML Tag. | |

| Entity Ancestry | | | |
|---|-----------------------------|---|------------------------|
| This table specifies the parent-child relationship between two entities in the conceptual model. For e.g., there is a parent-child relationship between POLICY & POLICY LOCATION. | | | |
| Field Name | Attribute Definition | Description | Possible Values |
| Id | N 15 | Unique identifier | |
| Parent-EntityId | FK | Identifies of the parent component of a parent-child/ hierarchical relationship | |
| Child-EntityId | FK | Identifies of the child component of a parent-child/ hierarchical relationship | |

| Question | | | |
|---|-----------------------------|--|---|
| This table specifies each 'Question' associated with an Entity. | | | |
| Field Name | Attribute Definition | Description | Possible Values |
| EntityId | FK | Identifies the Entity that this question belongs to. | |
| Seq | int | Identifies the question within the Entity. With the EntityId, it uniquely identifies a question. | |
| Source | T 1 | Defines how the question's answer is obtained | N, L, D, F |
| Category | T 1 | Determines how the question can be used. All values besides Other, have a specific meaning to the system that have different logic in the system associated with them. A Question of Type C designates that it is the code, L designates that it is the Literal, S designates it is the display sequence number, D designates the long description for the Code. These are used in building dropdowns. A Question of type "O" is used for all other usage. | C, L, S, D, O |
| Domain | T 3 | Describes the answer type for the question. | TXT-Text, INT-Integer, REL-Real, IND-Indicator, DAT-Date, ZIP-Zip, FIL-File |
| Standard Reference Code | T 20 | Standard Tag that is associated with the Question. An example could be the ACCORD XML Tag. | |
| Storage Code | T 1 | Indicates if the answer should be stored. Values Include: Stored, Not Stored, Stored If different. "Stored if different" specifies that the answer for the question is created only if the entered value is different from default. | S, N, D |
| Default Value | T 50 | Default Answer that should be shown when the question is asked. The field will be set with this value and saved if the field is hidden when the details are added and submitted | |

NOT FURNISHED UPON FILING

| Contract Question By Product | | | |
|---|-----------------------------|--|--|
| This table specifies the properties for a Contract Question of a Product. | | | |
| Field Name | Attribute Definition | Description | Possible Values |
| Id | N 15 | Unique identifier | |
| EntityId | FK | Identifies the Entity for the Question | |
| QuestionSeq | FK | Identifies the question | |
| ProductId | FK | Identifies which product this question belongs | |
| Standard Reference Code | T 50 | Standard Tag that is associated with the Question. An example could be the ACCORD XML Tag. This overrides the value in QUESTION.. | |
| Answer Option Category | T 1 | Determines what types of answers options are possible (None, specific list, or range). This overrides the value in QUESTION. | 0- No specific list of values or range 1- The possible values are specified in Answer Option or Product Answer Option, each value in a separate record 2- The range is specified using 'Range From', 'Range To' and 'Increment' in one record of the Answer Option or Product Answer Option. |
| Source | T 1 | Defines how the question's answer is obtained. This overrides the value in QUESTION. | N, L, D, F |
| Storage Code | T 1 | Indicates if the answer should be stored. Values Include: Stored, Not Stored, Stored If different. This overrides the value in QUESTION. | S, N, D |
| Presentation Literal | T 50 | Text that is displayed on the screen for this question (the Label). This overrides the value in QUESTION. | |

| Question Relation | | | |
|--|------|---|------------|
| This table specifies the relationship between two Questions. | | | |
| Id | N 15 | Unique identifier | |
| ProductId | FK | Identifies which product this question relation belongs. | |
| EntityId | FK | Identifies the Entity for the Question that the relation is defined for | |
| QuestionSeq | FK | Identifies the question that the relation is defined for | |
| Type | T 1 | Type of Relation | A, L, S, V |
| Target EntityId | FK | Identifies the Entity of the target Question | |
| Target QuestionSeq | FK | Identifies the target Question | |
| Ref EntityId | FK | Identifies the Entity of the reference Question | |
| Ref QuestionSeq | FK | Identifies the reference Question | |

View Item

| This table defines each item in a View. The View Item can be an Image or a Literal (text) or a Question (label and control for data entry). The View Item specifies the presentation characteristics for each displayed element | | | |
|---|----------------------|---|--|
| Field Name | Attribute Definition | Description | Possible Values |
| Id | N 15 | Unique identifier | |
| Seq No | Int 3 | Determines the order of appearance as well as the tab order of the fields on the screen. | |
| View-Tag | FK | Identifies which View this Item belongs | |
| EntityId | FK | Identifies the Entity for the Question for the View Item | |
| QuestionSeq | FK | Identifies the question for the View Item | |
| ProductId | FK | Identifies which product this view item is associated to | |
| Type | T 1 | Identifies the type of the view item. Image, Literal, Question | I - Image L - Literal Q - Question |
| Presentation Literal | T 1000 | This specifies the label displayed for the Question or the Literal to be displayed. | |
| Banner Width | Int 4 | Identifies the width of the column in pixels for a List view. | |
| Protection Code | T 2 | Show, Not protect | SN |
| | | Show, Protected | SP |
| | | Show, Protected on Change | SC |
| | | Show when populated not protect | CN (previously DN) |
| | | Show when populated protected | CP (previously DP) |
| | | Hide (use default if applicable) | HD |
| | | Hide, set to "missing" on Add | HM |
| Display Code | T 1 | Describes how an answer (or list of answers) should display itself on the presentation. Values include Code, Literal, and BothC - the field will be populated with the Code (e.g NY)L - the field will be populated with the Literal value (e.g. New York)B - the field will be populated with the concatenated value of the Code and the Literal (e.g. "NY New York") | C, L, B |

| | | | |
|------------------------------|-------|--|---|
| Segment | Int 3 | Identifies what segment this question belongs to. Often, depending on the responses to a question, a different set of questions may need to be asked or the possible answer options may be different. In such cases the Detail Section should be organized into one with multiple segments. Instead of displaying all the questions at once, the questions will be displayed segment by segment in succession based on the responses to the questions. | |
| Length | Int 3 | Defines the length of the viewable area of the answer (how much of the answer can be seen). | |
| Max Length | Int 4 | Maximum length the field can have (the length is typically smaller...) | |
| Default Value | T50 | If present, it specifies the overriding value for the Default value specified in QUESTION table for the question. | |
| Text Area Height | Int 3 | The number of viewable lines for the a text area | |
| Before Decimal Length | Int 2 | If it is a numeric field, defines the maximum number of digits before the decimal point (integer part) | |
| After Decimal Length | Int 2 | If it is a numeric field, defines the maximum number of digits after the decimal point (decimal part) | |
| Required Ind | T 1 | If it is Y, the field is a mandatory field. The system will not permit the user to submit the page until a valid value is entered in this field. | Y, N |
| Literal Start Row | Int 2 | Identifies what row the literal for the question should be placed | |
| Literal Row Span | Int 1 | Identifis how many rows the literal should span | |
| Literal Start Column | Int 2 | Identifies what column the literal for the question should be placed. | 0 - 9 (The main body is divided into 10 columns.) |
| Literal Column Span | Int 2 | Identifies how many columns the literal should span. | 1 to 10 |
| Literal Horizontal Alignment | T 10 | Identifies the justification for the literal text within the cell. | Left, Right, Center |
| Literal Vertical Alignment | T 10 | Identifies the how the literal text should be aligned. | Top, Middle, Bottom |
| Question Start Row | Int 1 | Identifies what row the answer should be placed | |
| Question Row Span | Int 1 | Identifis how many rows the answer should span | |
| Question Start Column | Int 1 | Identifies what column the answer should be placed. | 1 to 10 |

| | | | |
|-------------------------------|-------|--|---|
| Question Column Span | Int 1 | Identifies how many columns the answer should span. | 1 to 10 |
| Question Horizontal Alignment | T 10 | Identifies the justification for the answer within the cell. | Left, Right, Center |
| Question Vertical Alignment | T 10 | Identifies the how the answer should be aligned. | Top, Middle, Bottom |
| Question Control Type | T 20 | Identifies what type of control should be used on the presentation to capture the answer. | COMBO, LIST, LITERAL, RADIO, CHECKBOX, TEXT, TEXTAREA |
| Question Format | T 20 | Identifies what format the numeric fields should have. DATE1 (###/###/####), ZIP4 (####-####), SS (### ## ####) COMMA (#,###), USCUR (\$###,###.##), PHONE (###) ###-####. Other types most likely will be added in the future | DATE1, ZIP4, SS, COMMA, USCUR, PHONE |
| Image Width | N 4 | The width in pixels of the image | |

View Item Dependency Line

This table defines the dependencies or the condition under which the the View Item (display element) is presented. This provides a way by which a display element can be presented only under certain conditions. Dependency Line along with Dependency Value enables specification of any complex expression. The Dependency Expression is a complex boolean expression with the left side operand is a Question and the right side operand can be a literal or a Question. The comparison operator can be one of <, >, <=, >=, <>, =. The operations can be grouped by appropriate parentheses. For example a valid expression is ((Coverage=C1 or C2 or C3) AND (Vehicle Type = T1 or T2 or T3)).

| Field Name | Attribute Definition | Description | Possible Values |
|---------------------|----------------------|---|---|
| ID | N 15 | Unique identifier for the instance | |
| View Item Id | FK | Identifies the View Item that is the subject of this dependency | |
| Seq-No | T 8 | Defines the order in which the expression is built. | |
| O-Paran-Cnt | Int 2 | # of open parenthesis that precede the line | |
| Primary EntityId | FK | Identifies the Entity of the Primary Question whose value will is used to evaluate the dependency | |
| Primary QuestionSeq | FK | Identifies the Primary Question whose value will is used to evaluate the dependency | |
| C-Operator | T 2 | Comparison Operator | <, > <=, >=, <>, =, ispresent, isnotpresent |
| C-Paran-Cnt | Int 2 | # of closed parentheeis that follow the line | |
| AndOr | T 3 | Identifies how this dependency line should be resolved with the next dependency line | AND, OR |

Dependency Value

This table defines the set of values that need to be satisfied with Dependency Line Primary Q. Each instance specifies one value of the OR expression (in the example C1, C2 and C3). Each value may be a constant or another Question (specified here as the Secondary Question)

| Field Name | Attribute Definition | Description | Possible Values |
|------------------------------------|----------------------|--|--|
| View Item Dependency Line Id | FK | Foreign key to Dependency Line | |
| Seq-No | Int 4 | Sequence number to identify the occurrence. Along with DL ID, it uniquely identifies the occurrence. | |
| Secondary EntityId | FK | Identifies the Entity of the Secondary Question whose value will be compared against the Primary Question in Dependency Line | |
| Secondary QuestionSeq | FK | Identifies the Secondary Question whose value will be compared against the Primary Question in Dependency Line | |
| Value | t 20 | Literal Value of the question that should be used in the determination of the dependency. | In the example above C1, C2, C3, T1, T2 and T3 are the literal values used for comparison. |

Master Product Occurrence

This table contains one record for every instance of a Product.

| Field Name | Attribute Definition | Description | Possible Values |
|------------|----------------------|------------------------------------|-----------------|
| ProductId | N 15 | Unique identifier for the Product. | |

Product Occurrence

This table contains one record for each occurrence of each entity in the Product hierarchy.

| Field Name | Attribute Definition | Description | Possible Values |
|------------|----------------------|---|-----------------|
| P-Id | N 15 | Unique identifier | |
| ProductId | FK | Identifies the Product that this Product occurrence belongs to. | |
| EntityId | FK | Identifies the Entity for the Occurrence | |

Product Ancestry

| Identifies the parent-child relationships between Product Occurrences. | | | |
|--|----------------------|--|-----------------|
| Field Name | Attribute Definition | Description | Possible Values |
| Id | N 15 | Unique identifier | |
| Parent-P-Id | FK | Product occurrence identifier of the parent. | |
| Child-P-Id | FK | Product occurrence identifier of the child. | |

Product Answer

| This table contains the values for the individual attributes of a Product Occurrence. | | | |
|---|----------------------|--|-----------------|
| Field Name | Attribute Definition | Description | Possible Values |
| PA-Id | N 15 | Unique identifier | |
| P-Id | FK | Identifies which Product Occurrence this answer is for | |
| EntityId | FK | Identifies the Entity for the Answer | |
| QuestionSeq | FK | Identifies the question for the Answer | |
| Q-Category | T 1 | (Denormalized field from the Question Table) | |
| Value | T 20 | Value of the Answer (if longer than 20, then the value is stored in Long Product Answer Value) | |
| AO id | FK | If the value of the Answer for the product side entity's attribute is a discrete value chosen from an Answer Option list, this field specifies the id of that answer option. | |
| User ID | T 20 | The logon user ID of the user who created this Answer | |
| Timestamp | Timestamp | The date and time this Answer was created | |

Long Product Answer

| This table contains the Product Answer value if the Product Answer's size exceeds 20 characters. | | | |
|--|----------------------|-------------------------------|-----------------|
| Field Name | Attribute Definition | Description | Possible Values |
| PA-Id | N 15 | Foreign key to Product Answer | |
| Value | T 1000 | Value of the Answer | |

Contract Question By Product Occurrence

This table specifies the overriding values for presentation characteristics of a Question based on Product Occurrence.

| Field Name | Attribute Definition | Description | Possible Values |
|------------------------|----------------------|--|--|
| PO-Id | N 15 | Unique identifier | |
| EntityId | FK | Identifies the Entity for the Question | |
| QuestionSeq | FK | Identifies the Question | |
| Answer Option Category | T 1 | Determines what types of answers options are possible (None, specific list, or range). | 0- No specific list of values or range 1- The possible values are specified in Product Answer Option, each value in a separate record 2- The range is specified using 'Range From', 'Range To' and 'Increment' in one record of the Product Answer Option. |
| Default Value | T 50 | This field should be always specified. It cannot be null/empty. It specifies the Default value for the contract question as applicable in a product. | |
| Presentation Literal | T 1000 | This field is normally empty/null. If a valid value is present it specifies the label displayed for the Question or the Literal to be displayed. | |
| Protection Code | T 2 | This field is normally empty/null. If a valid value is present it specifies Protection Code for the field as applicable to the product. (See View Item) | |
| Display Code | T 1 | This field is normally empty/null. If a valid value is present it specifies how an answer (or list of answers) should display itself on the presentation. (See View Item) | |
| Required Indicator | T 1 | This field is normally empty/null. If a valid value is present it specifies whether or not the field is a mandatory field. | Y, N |
| Question Control Type | T 20 | This field is normally empty/null. If a valid value is present it specifies the type of control that should be used on the presentation to capture the answer. (See View Item) | |
| Question Format | T 20 | This field is normally empty/null. If a valid value is present it specifies the format the field should have. (See View Item) | |

Product Answer Option

This table specifies the possible values for a the Answer of a Question relative to a particular Product Occurrence. Question. If the Product Occurrence Question Answer Option Category is 1 (specific list), each possible value is specified in this table, one record per value. If it is 2 (ranges of values), then each range must be specified in one record using From, To and Increment.

| Field Name | Attribute Definition | Description | Possible Values |
|-----------------|----------------------|---|-------------------|
| PAO-Id | N 15 | Unique identifier | |
| PO-Id | FK | Identifies which Product Occurrence Option this answer options belongs to | |
| AO-Id | FK | Identifier of the Answer Option this Product Answer Option is linked. | |
| Sequence Number | Int 4 | sets the order in which the answers should be displayed in a dropdown | |
| Value | T 20 | This field specifies the Value that is to be stored in the Contract Answer area | e.g. "NJ" |
| Literal | T 50 | This field specifies the literal text that needs to be displayed in the dropdown. | e.g. "New Jersey" |
| Description | T 1000 | Textual description for the answer. Usually used in help text areas | |
| From | Dec | Starting value of the allowed range of values. The range includes this value. | |
| To | Dec | Ending value of the allowed range of values. The range includes this value. | |
| Increment | Dec | The increment for the range of discrete values. | |

Answer Option (AO)

This table specifies the possible values for all Questions including variations by Product. If the Question Answer Option Category is 1 (specific list), each possible value is specified in this table, one record per value. If it is 2 (ranges of values), then each range must be specified in one record using From, To and Increment.

| Field Name | Attribute Definition | Description | Possible Values |
|-----------------|----------------------|---|-------------------|
| AO-Id | N 15 | Unique identifier | |
| Q-Tag | FK | Identifies which question (B-Tag and Q-Seq) this answer option belongs to | |
| Sequence Number | Int 4 | sets the order in which the answers should be displayed in the dropdown | |
| Value | T 20 | Value that is to be stored in the Contract or Product Answer | e.g. "NJ" |
| Literal | T 50 | Textual answer that is displayed in the dropdown boxes for the answer | e.g. "New Jersey" |
| Description | T 1000 | Textual description for the answer. Usually used in help text areas | |
| From | Dec | Starting value of the allowed range of values. The range includes this value. | |
| To | Dec | Ending value of the allowed range of values. The range includes this value. | |
| Increment | Dec | The increment for the range of discrete values. | |

Master Contract Occurrence

This table contains one record for every instance of a Contract. This record ties together all the details for the Contract.

| Field Name | Attribute Definition | Description | Possible Values |
|------------|----------------------|-------------------------------------|-----------------|
| ContractID | N 15 | Unique identifier for the Contract. | |

Contract Occurrence (C)

This table contains one record for each occurrence of each entity in the Contract hierarchy.

| Field Name | Attribute Definition | Description | Possible Values |
|------------|----------------------|---|-----------------|
| C-Id | N 15 | Unique identifier for the policy side entity type occurrence. | |
| ContractID | FK | Identifies the Contract this Contract Occurrence belongs to | |
| EntityId | FK | Identifies the Entity for the Occurrence | |

Contract Ancestry

Identifies the parent-child relationships between Contract Occurrences.

| Field Name | Attribute Definition | Description | Possible Values |
|------------|----------------------|--|-----------------|
| Id | N 15 | Unique identifier | |
| Parent-Id | FK | Contract occurrence identifier of the parent | |
| Child-Id | FK | Contract occurrence identifier of the child | |

Contract Answer (CA)

This table contains the values for the individual attributes of a Contract Occurrence.

| Field Name | Attribute Definition | Description | Possible Values |
|-------------|----------------------|--|-----------------|
| CA-Id | N 15 | Unique identifier | |
| C-Id | FK | Identifies which Contract Occurrence this answer is for. Each logical record of every policy side entity type is identified by this ID. | |
| EntityId | FK | Identifies the Entity for the Answer | |
| QuestionSeq | FK | Identifies the question for the Answer | |
| Value | T 20 | Value of the Answer (if longer than 20, then the value is stored in Long Contract Answer Value) | |
| PAO ID | FK | If the value of the Answer for the policy side entity's attribute is a discrete value chosen from a Product Answer Option list, this field specifies the id of that answer option. | |
| User ID | T 20 | The logon user ID of the user who created this Answer | |
| Timestamp | Timestamp | The date and time this Answer was created | |

Long Contract Answer

This table contains the Contract Answer value if the Contract Answer's size exceeds 20 characters

| Field Name | Attribute Definition | Description | Possible Values |
|------------|----------------------|-------------------------------|-----------------|
| CA-Id | N 15 | Foreign key to Product Answer | |
| Value | T1000 | Value of the Answer | |

WHAT IS CLAIMED IS:

1. A computer readable media having a plurality of programming instructions encoded thereon an adaptive software application including a plurality of several different types of programming modules, each of said programming modules being highly parameterized and to receive a plurality of parameters from an external source that governs a behavior of said each programming module to capture information, display information and store said captured information in one or more occurrence databases, whereby each of said plurality of modules is adaptable to varying business requirements through said plurality of externally provided parameters, and said adaptive software application is assemblable through repeated copying of various combinations of said plurality of several different types of programming modules.

2. The computer readable media according to claim 1, wherein each of the plurality of several different types of programming modules includes:

an interface component to present information to and accept information from one or more outside sources;

a processing logic component to manipulate and evaluate information based on received parameters; and

a data persistence logic component to retrieve, add, update and delete data targeting one or more occurrence databases.

3. The computer readable media according to claim 2, wherein said processing logic component performs comparisons, computations, data association and data dependency decisions.

4. The computer readable media according to claim 2, wherein said one or more outside sources includes a user interface.

5. The computer readable media according to claim 2, wherein said one or more outside sources includes a World Wide Web page.

6. The computer readable media according to claim 2, wherein said one or more outside sources includes an application programming interface.

7. The computer readable media according to claim 2, wherein said interface component is parameter driven.

8. The computer readable media according to claim 2, wherein said processing logic component is parameter driven.

9. The computer readable media according to claim 2, wherein said data persistence component is parameter driven.

10. The computer readable media according to claim 2, wherein said interface component receives one or more parameters from a definitional database storing information to control behavior of the interface component to display information and capture information.

11. The computer readable media according to claim 2, wherein said processing logic component receives one or more parameters from a definitional database storing information controlling behavior of the processing logic component to manipulate data.

12. The computer readable media according to claim 2, wherein said data persistence component receives one or more parameters from a definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database.

13. The computer readable media according to claim 2, wherein said interface component receives one or more parameters from a definitional database storing information to control behavior of the interface component to display information and capture information, and said data processing component receives one or more parameters from the definitional database to control behavior of the processing logic component to manipulate data.

14. The computer readable media according to claim 2, wherein said processing logic component receives one or more parameters from a definitional database to control behavior of the processing logic component to manipulate data, and said data persistence component receives one or more parameters from the definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database.

15. The computer readable media according to claim 2, wherein said data persistence component receives one or more parameters from a definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database, and said interface component receives one or more parameters from the definitional database storing information to control behavior of the interface component to display information and capture information.

16. The computer readable media according to claim 2, wherein said data persistence component receives one or more parameters from a definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database, said interface component receives one or more parameters from the definitional database storing information to control behavior of the interface component to display information and capture information, and said processing logic component receives one or more parameters from the definitional database to control behavior of the processing logic component to manipulate data.

17. The computer readable media according to claim 2, wherein said instructions further comprise one or more occurrence databases storing information captured by one or more of the plurality of several different types of programming modules.

18. The computer readable media according to claim 17, wherein each of the one or more occurrence databases storing occurrence information is assigned to one business process supported by a business application for which the adaptive software application is being adapted.

19. The computer readable media according to claim 2, wherein each of said one or more occurrence databases storing occurrence information includes one or more generic tables.

20. The computer readable media according to claim 19, wherein the one or more generic tables are not specific to any business process.

21. The computer readable media according to claim 2, wherein each of the one or more occurrence databases storing occurrence information includes one or more meta-model driven tables.

22. The computer readable media according to claim 2, wherein the one or more meta-model driven tables are not specific to any business process.

23. An apparatus that can be adapted to a wide variety of business applications comprising:

a first database storing a plurality of definitional parameters;

one or more second databases, each of said one or more second databases including a plurality of generic tables;

a processor coupled to said first database and said one or more second databases; and

a computer readable media having encoded thereon an adaptive application including a plurality of different types of programming modules, each of said programming modules being highly parameterized and to receive a plurality of parameters from a first database that govern a behavior of said each programming module to capture information, display information and store said captured information in said one or more second databases, whereby each of said plurality of modules is adaptable to varying business requirements through said plurality of parameters stored in the first database, and said adaptive software application is assemblable through repeated copying of various combinations of said plurality of several different types of programming modules.

24. The apparatus according to claim 23, wherein each of the plurality of several different types of programming modules includes:

an interface component to present information to and accept information from one or more outside sources;

a processing logic component to manipulate and evaluate information based on parameters received from the first database; and

a data persistence logic component to retrieve, add, update, and delete data targeting said one or more second databases.

25. The apparatus according to claim 24, wherein said processing logic component performs comparisons and data dependency decisions.

26. The apparatus according to claim 24, wherein said one or more outside sources includes a user.

27. The apparatus according to claim 24, wherein said one or more outside sources includes a user interacting with the interface component via a World Wide Web page.

28. The apparatus according to claim 24, wherein said one or more outside sources includes an application programming interface.

29. The apparatus according to claim 24, wherein said interface component is parameter driven.

30. The apparatus according to claim 24, wherein said processing logic component is parameter driven.

31. The apparatus according to claim 24, wherein said data persistence component is parameter driven.

32. The apparatus according to claim 24, wherein said interface component receives one or more parameters from the first database.

33. The apparatus according to claim 24, wherein said processing logic component receives one or more parameters from the first database.

34. The apparatus according to claim 24, wherein said data persistence component receives one or more parameters from the first database.

35. The apparatus according to claim 24, wherein said interface component and said processing logic component receive one or more parameters from the first database.

36. The apparatus according to claim 24, wherein said processing logic component and said data persistence component receive one or more parameters from the first database.

37. The apparatus according to claim 24, wherein said data persistence component and said interface component receive one or more parameters from the first database.

38. The apparatus according to claim 24, wherein said data persistence component, said processing logic component and said interface component receive one or more parameters from the first database.

39. The apparatus according to claim 24, wherein each of the one or more second databases is assigned to one business process supported by a business application for which the adaptive application is being adapted.

40. The apparatus according to claim 24, wherein each of said one or more second databases includes one or more generic tables.

41. The apparatus according to claim 40, wherein the one or more generic tables are not specific to any business process.

42. The apparatus according to claim 24, wherein each of the one or more the second databases includes one or more meta-model driven tables.

43. The apparatus according to claim 24, wherein the one or more meta-model driven tables are not specific to any business process.

44. A computer readable media have encoded thereon a software application that can be adapted to varying business requirements, said software application comprising a plurality of adaptive units adapted to interact with a user and further being programmed to:

- utilize data from a definition database as parameters that govern operations for capturing, manipulating and storing of information;

- store and retrieve information in an occurrence database including meta-model based tables; and

- accommodate storing new data fields in the occurrence database without requiring restructuring said occurrence database.

45. A method for adapting a software application to varying business requirements comprising:

- providing a plurality of adaptive units adapted to interact with a user, perform data manipulation functions, and data persistence functions, said adaptive units:

- utilizing data from an external source as parameters that govern operations for capturing, manipulating and storing information;

- storing and retrieving information in an occurrence database including meta-model based tables; and

- accommodating storage of new data fields in the occurrence database without requiring restructuring said occurrence database.

46. A method for creating a software application comprising:
creating one or more web pages for each of a plurality of previously identified user interface screens for the software application by copying a plurality of preexisting web page element templates in various combinations;
assigning one or more view names to each web page element template copied;
inserting the one or more view names into each copied web page element template; and
using the one or more assigned view names to obtain a plurality of parameters for the web page element template from a definition database.

47. The method according to claim 46, further comprising storing in a definition database all necessary information associated with each assigned view name.

48. The method according to claim 47, wherein all necessary information includes one or more of the following: a view, an entity, an entity ancestry, a plurality of questions, a plurality of question relations, a plurality of answer options, a plurality of view items, a plurality of view item dependency lines, and a plurality of view item dependency values.

49. A method for creating software for a business application comprising:
providing a database including at least a definition portion and an occurrence portion;
providing a plurality of adaptive units that when arranged in various combinations create a user interface; and
providing a plurality of preexisting code components, which are reusable and modular in form.

50. A method for creating software for a business comprising:
storing in a definition database information about the business that is common to one or more processes in a linear flow of the business;

storing in one or more meta-model based tables in each of one or more occurrence databases information related to a particular business process of the one or more processes; and

controlling a behavior of one or more adaptive program units to display information, capture information and store captured information in the one or more occurrence databases using one or more parameters from information in the definition database.

51. The method according to claim 50, further comprising linking data in the one or more occurrence databases of one or more subsequent processes of the one or more processes of the business to one or more prior processes in the one or more processes to provide details in the one or more subsequent processes that are determinable by the one or more prior processes.

52. A computer readable media having encoded thereon an adaptive software application comprising:

a plurality of reusable elements, each of said reusable elements being highly parameterized and to receive a plurality of parameters from an external source that governs a behavior of said each reusable element to capture information, display information and store said captured information in one or more occurrence databases, whereby each of said plurality of reusable elements is adaptable to varying business requirements through said plurality of externally provided parameters, and said adaptive software application is assemblable through repeated copying of various combinations of said plurality of reusable elements, each of the plurality of reusable elements including:

an interface component to present information to and accept information from one or more outside sources;

a processing logic component to manipulate and evaluate information based on received parameters; and

a data persistence logic component to retrieve, add, update and delete data targeting one or more occurrence databases.

53. The computer readable media according to claim 52, wherein the plurality of reusable elements includes a list adaptive unit.

54. The computer readable media according to claim 52, wherein the plurality of reusable elements includes a detail adaptive unit.

55. The computer readable media according to claim 52, wherein the plurality of reusable elements includes an updateable list adaptive unit.

56. The computer readable media according to claim 52, wherein the plurality of reusable elements includes a dual list adaptive unit.

57. The computer readable media according to claim 52, wherein the plurality of reusable elements includes a list adaptive unit and a detail adaptive unit.

58. The computer readable media according to claim 53, wherein the list adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, a view/edit event, an add event and a delete event.

59. The computer readable media according to claim 58, wherein the list adaptive unit includes one or more of the following properties: a ViewName property, a DetailTarget property, a ProductId property and an Occurrence ID property.

60. The computer readable media according to claim 54, wherein the detail adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, a new event, a continue event and a submit event.

61. The computer readable media according to claim 60, wherein the detail adaptive unit includes one or more of the following properties: a ViewName property, an Occurrence ID property, a ProductId property and a SegmentNo property.

62. The computer readable media according to claim 56, wherein the dual-list adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, a view/edit event, an “add >>” event, and a “<<Remove” event.

63. The computer readable media according to claim 62, wherein the dual-list adaptive unit includes one or more of the following properties: a LeftViewName property, RightViewName property, a DetailTarget property, a LinkedEntity property, a LinkedQuestion property, an Occurrence ID property and a ProductId property.

64. The computer readable media according to claim 55, wherein the updateable-list adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, an add row event, a delete row event, and a submit event.

65. The computer readable media according to claim 64, wherein the updateable-list adaptive unit includes one or more of the following properties: a ListViewName property, a DetailViewName property, an Occurrence ID property, and a ProductId property.

66. The computer readable media according to claim 57, wherein the list adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, a view/edit event, an add event and a delete event, the list adaptive unit includes one or more of the following properties: a ViewName property, a DetailTarget property, a ProductId

property and an Occurrence ID property, and initiating the add event in the list adaptive unit activates a new event of a detail adaptive unit using the DetailTarget property, the ProductId property, and the OccurrenceId property of a parent occurrence for a list.

67. The computer readable media according to claim 59, wherein initiating the delete event of the list adaptive unit deletes the contract occurrence with the occurrenceID property and the deleted occurrence is removed from a display.

68. The computer readable media according to claim 57, wherein the list adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, a view/edit event, an add event and a delete event, the list adaptive unit includes one or more of the following properties: a ViewName property, a DetailTarget property, a ProductId property and an Occurrence ID property, and initiating the view/edit event in the list adaptive unit activates the load event of the detail adaptive unit using the DetailTarget property, the ProductId property, and the OccurrenceId property of a selected item.

69. The computer readable media according to claim 59, wherein initiating the load event in the list adaptive unit comprises the following actions:

- a plurality of View attributes are retrieved using the ViewName property;
- a plurality of attributes of each View Item are retrieved for the ViewName property and the ProductId property;
- for each View Item, a plurality of Question attributes are retrieved;
- using an OccurrenceId property as a "parent-id", a plurality of "children" Contract Occurrences are retrieved from a Contract Ancestry;
- a plurality of Contract Answers are retrieved for a plurality of Questions related to the View Items from each occurrence; and
- a plurality of answers are displayed for each Contract Occurrence in a separate line.

70. The computer readable media according to claim 61, wherein initiating the load event in the detail adaptive unit comprises the following actions:

- a) initializing the SegmentNo to one;
- b) retrieving a plurality of View Attributes using the ViewName;
- c) retrieving a plurality of attributes for each View Item for the ViewName and the ProductId;
- d) retrieving for each View Item, a plurality of View Item Dependency Lines and a plurality of View Item Dependency Values;
- e) performing, for each View Item related to a Question, the following:
 - (i) retrieving a plurality of question attributes; and
 - (ii) performing, if the SegmentNo is equal to the a Segment of the View Item, the following:
 - (1) if the ProductId is not zero, overriding a plurality of Question attributes using a Contract Question by Product;
 - (2) retrieving, if the question has a plurality of answer options and ProductID is zero, a plurality of attributes of each answer option;
 - (3) using the OccurrenceID retrieving the contract answer, if any, for a question; and
 - (4) performing, if the question has a plurality of question relations, the following:
 - (A) retrieving a plurality of attributes of each question relation;
 - (B) retrieving, for a default relation and a derive relation, a Contract Answer from a plurality of higher level contract entities using the OccurrenceId and a Contract Ancestry to determine a value of an answer;
 - (C) retrieving, for a link question relation, a set of possible selections using the ProductId; and
 - (D) retrieving, for a dynamic question relation, a plurality of attributes from a Contract Question By Product Occurrence and from a Contract Answer Option by Product Occurrence; and

f) displaying, after eliminating a plurality of View Items that are inapplicable because of a plurality of question dependency conditions, a remaining plurality of View Items with a Segment equal to the SegmentNo along with a plurality of answers that are not hyperlinked.

71. The computer readable media according to claim 61, wherein initiating the new event in the detail adaptive unit comprises the following actions:

- a) initializing the SegmentNo to one;
- b) retrieving a plurality of View Attributes using the ViewName;
- c) retrieving a plurality of attributes for each View Item for the ViewName and the ProductId;
- d) retrieving, for each View Item, a plurality of View Item Dependency Lines and a plurality of View Item Dependency Values;
- e) performing, for each View Item related to a Question, the following:
 - (i) retrieving a plurality of question attributes; and
 - (ii) performing, if the SegmentNo is equal to a Segment of the View Item, the following:
 - (1) if the ProductId is not zero, overriding a plurality of Question attributes using a Contract Question by Product;
 - (2) retrieving, if the question has a plurality of answer options and ProductID is zero, a plurality of attributes of each answer option;
 - (3) performing, if the question has a plurality of question relations, the following:
 - (A) retrieving a plurality of attributes of each question relation;
 - (B) retrieving, for a default relation and a derive relation, a Contract Answer from a plurality of higher level contract entities using the OccurrenceId and a Contract Ancestry to determine a value of an answer;
 - (C) retrieving, for a link question relation, a set of possible selections using the ProductId; and

(D) retrieving, for a dynamic question relation, a plurality of attributes from a Contract Question By Product Occurrence and from a Contract Answer Option by Product Occurrence; and
f) displaying, after eliminating a plurality of View Items that are inapplicable because of a plurality of question dependency conditions, a remaining plurality of View Items with a Segment equal to the SegmentNo along with a plurality of answers that are not hyperlinked.

72. The computer readable media according to claim 61, wherein initiating the continue event in the detail adaptive unit comprises the following actions:

- a) increasing the SegmentNo by one;
- b) performing, for each View Item with a Segment equal to the SegmentNo that is related to a Question, the following:
 - (i) if the ProductId is not zero, overriding a plurality of Question attributes using a Contract Question by Product;
 - (ii) retrieving, if the question has a plurality of answer options and ProductID is zero, a plurality of attributes of each answer option;
 - (iii) retrieving, using the occurrenceID, a contract answer, if any, for a question;
 - (iv) performing, if the question has a plurality of question relations, the following:
 - (1) retrieving a plurality of attributes of each question relation;
 - (2) retrieving, for a default relation and a derive relation, a Contract Answer from a plurality of higher level contract entities using the OccurrenceId and a Contract Ancestry to determine a value of an answer;
 - (3) retrieving, for a link question relation, a set of possible selections using the ProductId; and
- (D) retrieving, for a dynamic question relation, a plurality of attributes from a Contract Question By Product Occurrence and from a Contract Answer Option by Product Occurrence;

f) displaying the plurality of the View Items with a Segment less than SegmentNo along with a plurality of hyperlinked answers; and

g) displaying, after eliminating a plurality of View Items that are inapplicable because of a plurality of question dependency conditions, a remaining plurality of View Items with a Segment equal to the SegmentNo along with a plurality of answers that are not hyperlinked.

73. The computer readable media according to claim 61, wherein initiating the submit event in the detail adaptive unit comprises the following actions:

adding, if a new Contract Occurrence has been introduced, the new Contract Occurrence to a database;
adding each new Contract Answer to the database;
updating each Contract Answer that has been modified in the database; and
deleting one or more questions that have been deleted or made inapplicable by one or more dependency conditions from the database.

74. The computer readable media according to claim 63, wherein initiating the load event in the dual-list adaptive unit comprises the following actions:

a) performing, for a list on a right side and for selected items, the following:
(i) retrieving a plurality of View attributes using a RightViewName;
(ii) retrieving a plurality of attributes of each View Item for the RightViewName and the ProductId;
(iii) retrieving, for each View Item, a plurality of Question attributes;
(iv) using the OccurrenceId as a parent-id, a plurality of children Contract Occurrences are retrieved from a Contract Ancestry;
(v) obtaining a plurality of Contract Answers for the plurality of Questions related to the plurality of View Items from each occurrence; and
(vi) displaying the plurality of answers for each Contract Occurrence in a separate line; and
b) performing, for a list on a left side and for selected items, the following:
(i) retrieving a plurality of View attributes using a LeftViewName;

- (ii) retrieving a plurality of attributes of each View Item for the LeftViewName;
- (iii) retrieving, for each View Item, a plurality of Question attributes;
- (iv) finding all Product Occurrences that contain a Product Answer for the question identified by TargetEntity and TargetQuestion;
- (v) obtaining a plurality of Contract Answers for the plurality of Questions related to the plurality of View Items from each occurrence;
- (vi) displaying a plurality of answers for each Product Occurrence in a separate line excluding a plurality of occurrences that are already selected.

75. The computer readable media according to claim 52, wherein the plurality of reusable elements includes a dual-list adaptive unit and a detail adaptive unit.

76. The computer readable media according to claim 75, wherein the dual-list adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, a view/edit event, an “add >>” event, and a “<<Remove” event.

77. The computer readable media according to claim 76, wherein the dual-list adaptive unit includes one or more of the following properties: a LeftViewName property, RightViewName property, a DetailTarget property, a LinkedEntity property, a LinkedQuestion property, an Occurrence ID property and a ProductId property.

78. The computer readable media according to claim 76, wherein the detail adaptive unit includes one or more of the links associated with one or more of the events that can be initiated by clicking on the link associated with said one or more events, wherein said one or more events includes one or more of the following: a load event, a new event, a continue event and a submit event.

79. The computer readable media according to claim 78, wherein the detail adaptive unit includes one or more of the following properties: a ViewName property, an Occurrence ID property, a ProductId property and a SegmentNo property.

80. The computer readable media according to claim 79, wherein initiating the view/edit event in the dual-list adaptive unit activates the load event of the detail Adaptive Unit using the DetailTarget property, the ProductId property, and the OccurrenceId property of a selected item.

81. The computer readable media according to claim 79, wherein initiating the “add>>” event in the dual-list adaptive unit activates the new event of the detail Adaptive Unit using the DetailTarget property, the ProductId property, and the OccurrenceId property of a parent occurrence for a list.

82. The computer readable media according to claim 63, wherein initiating the “<<remove” event of the dual-list adaptive unit deletes the contract occurrence with the occurrenceID property, the deleted occurrence is removed from a display on the right list and a corresponding occurrence on a left list is displayed.

83. The computer readable media according to claim 65, wherein initiating the load event in the updateable adaptive unit comprises the following actions:

- a) retrieving a plurality of View Attributes using a ListViewName;
- b) retrieving a plurality of attributes for each View Item for the ListViewName and a ProductId;
- c) retrieving a plurality of attributes for each View Item for a DetailViewName and the productID;
- d) performing, for each View Item for the DetailViewName and the ProductId, the following:
 - (i) retrieving a plurality of question attributes;
 - (ii) overriding, the productId is not zero, the plurality of question attributes using a contract by question by product;

- (iii) retrieving, if the question has a plurality of answer options and ProductID is zero, a plurality of attributes of each answer option;
- (iv) performing, if the question has a plurality of question relations, the following:
 - (1) retrieving a plurality of attributes of each question relation;
 - (2) retrieving, for a default relation and a derive relation, a Contract Answer from a plurality of higher level contract entities using the OccurrenceId and a Contract Ancestry to determine a value of an answer;
 - (3) retrieving, for a link question relation, a set of possible selections using the ProductId; and
 - (4) retrieving, for a dynamic question relation, a plurality of attributes from a Contract Question By Product Occurrence and from a Contract Answer Option by Product Occurrence;
- f) using an OccurrenceId as a parent-id, retrieving a plurality of children Contract Occurrences from a Contract Ancestry;
- g) obtaining a plurality of Contract Answers for a plurality of Questions related to the DetailViewName View Items from each occurrence; and
- h) displaying a plurality of answers for each contract occurrence in a separate line.

84. The computer readable media according to claim 65, wherein initiating the add row event of the updateable list adaptive unit adds a blank new row to an updateable list filling in a plurality of default values, if any with no database changes.

85. The computer readable media according to claim 65, wherein initiating the delete row event of the updateable list adaptive unit deletes the indicated row from an updateable list with no database changes.

86. The computer readable media according to claim 65, wherein initiating the submit event in the updateable list adaptive unit comprises the following actions:
adding each new Contract Occurrence, if any, to a database;

adding each new Contract Answer, if any, to the database;
updating each Contract Answer that has been modified in the database; and
deleting one or more questions that have been deleted or made inapplicable by one or more dependency conditions from the database.

87. A computer readable media having encoded thereon an adaptive programming module that is highly parameterized receives a plurality of parameters from an external source that governs a behavior of said programming module to capture information, display information and store said captured information in an occurrence database, whereby said module is adaptable to varying business requirements through said plurality of externally provided parameters, wherein said adaptive programming module includes:

an interface component to present information to and accept information from an outside source;

a processing logic component to manipulate and evaluate information based on received parameters; and

a data persistence logic component to retrieve, add, update and delete data targeting the occurrence database.

88. The computer readable media according to claim 87, wherein said processing logic component performs comparisons and data dependency decisions.

89. The computer readable media according to claim 87, wherein said outside source includes a user interface.

90. The computer readable media according to claim 87, wherein said outside source includes a World Wide Web page.

91. The computer readable media according to claim 87, wherein said outside source includes an application programming interface.

92. The computer readable media according to claim 87, wherein said interface component receives one or more parameters from a definitional database storing information to control behavior of the interface component to display information and capture information.

93. The computer readable media according to claim 87, wherein said processing logic component receives one or more parameters from a definitional database storing information controlling behavior of the processing logic component to manipulate data.

94. The computer readable media according to claim 87, wherein said data persistence component receives one or more parameters from a definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database.

95. The computer readable media according to claim 87, wherein said interface component receives one or more parameters from a definitional database storing information to control behavior of the interface component to display information and capture information, and said data processing component receives one or more parameters from the definitional database to control behavior of the processing logic component to manipulate data.

96. The computer readable media according to claim 87, wherein said processing logic component receives one or more parameters from a definitional database to control behavior of the processing logic component to manipulate data, and said data persistence component receives one or more parameters from the definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database.

97. The computer readable media according to claim 87, wherein said data persistence component receives one or more parameters from a definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database, and said interface component receives

one or more parameters from the definitional database storing information to control behavior of the interface component to display information and capture information.

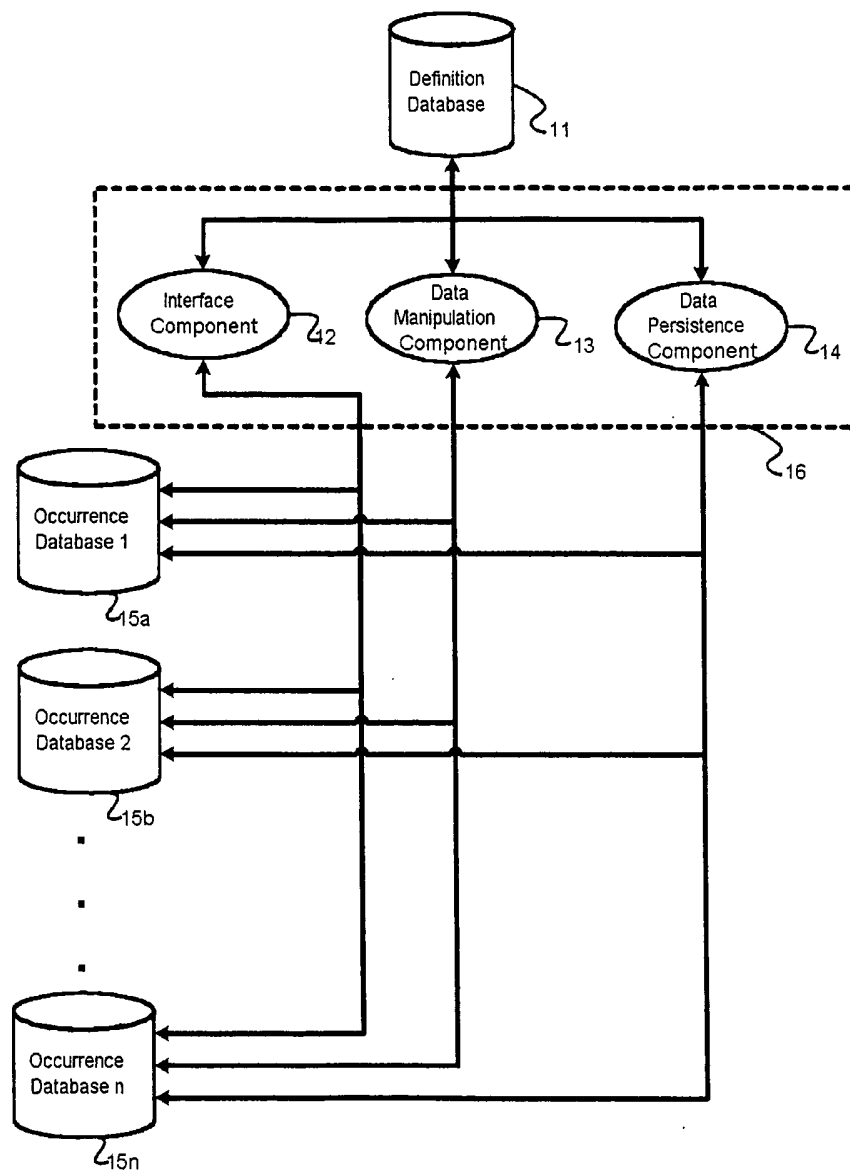
98. The computer readable media according to claim 87, wherein said data persistence component receives one or more parameters from a definitional database storing information controlling behavior of the data persistence component to store and edit information in a meta-model based database, said interface component receives one or more parameters from the definitional database storing information to control behavior of the interface component to display information and capture information, and said processing logic component receives one or more parameters from the definitional database to control behavior of the processing logic component to manipulate data.

99. The computer readable media according to claim 87, wherein the adaptive programming module comprises a list adaptive unit.

100. The computer readable media according to claim 87, wherein the adaptive programming module comprises a detail adaptive unit.

101. The computer readable media according to claim 87, wherein the adaptive programming module comprises an updateable list adaptive unit.

102. The computer readable media according to claim 87, wherein the adaptive programming module comprises a dual list adaptive unit.

**FIG 1**

Adaptive Application Data Structure
With a Single Business Process

20

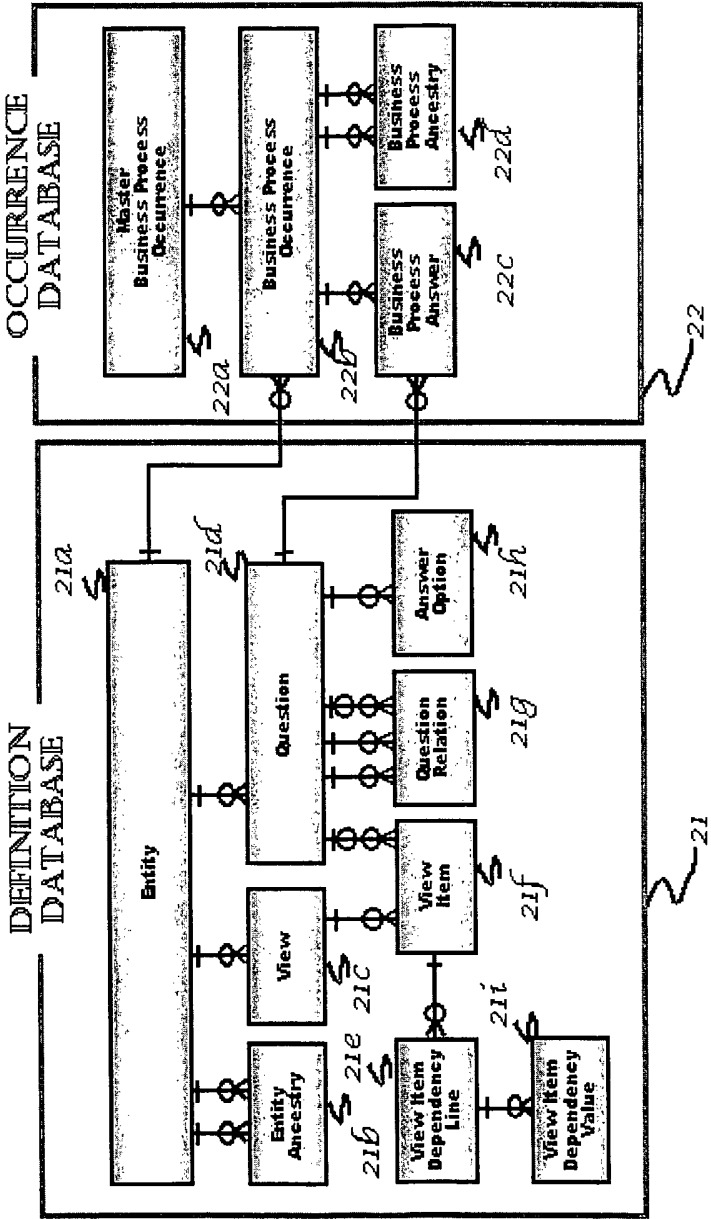


FIG 2

Adaptive Application Data Structure
With Multiple Business Processes

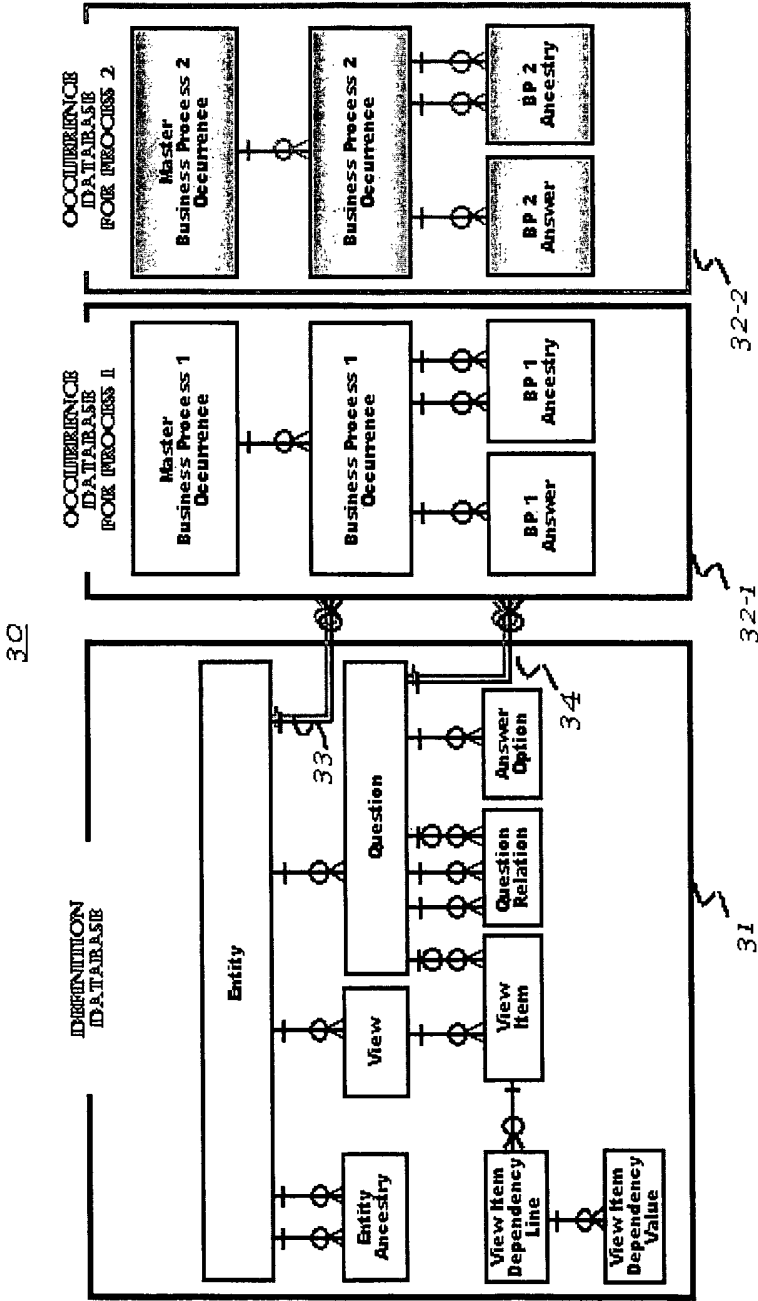


FIG 3

Adaptive Application Data Structure With Multiple Business Processes & Relative Customization

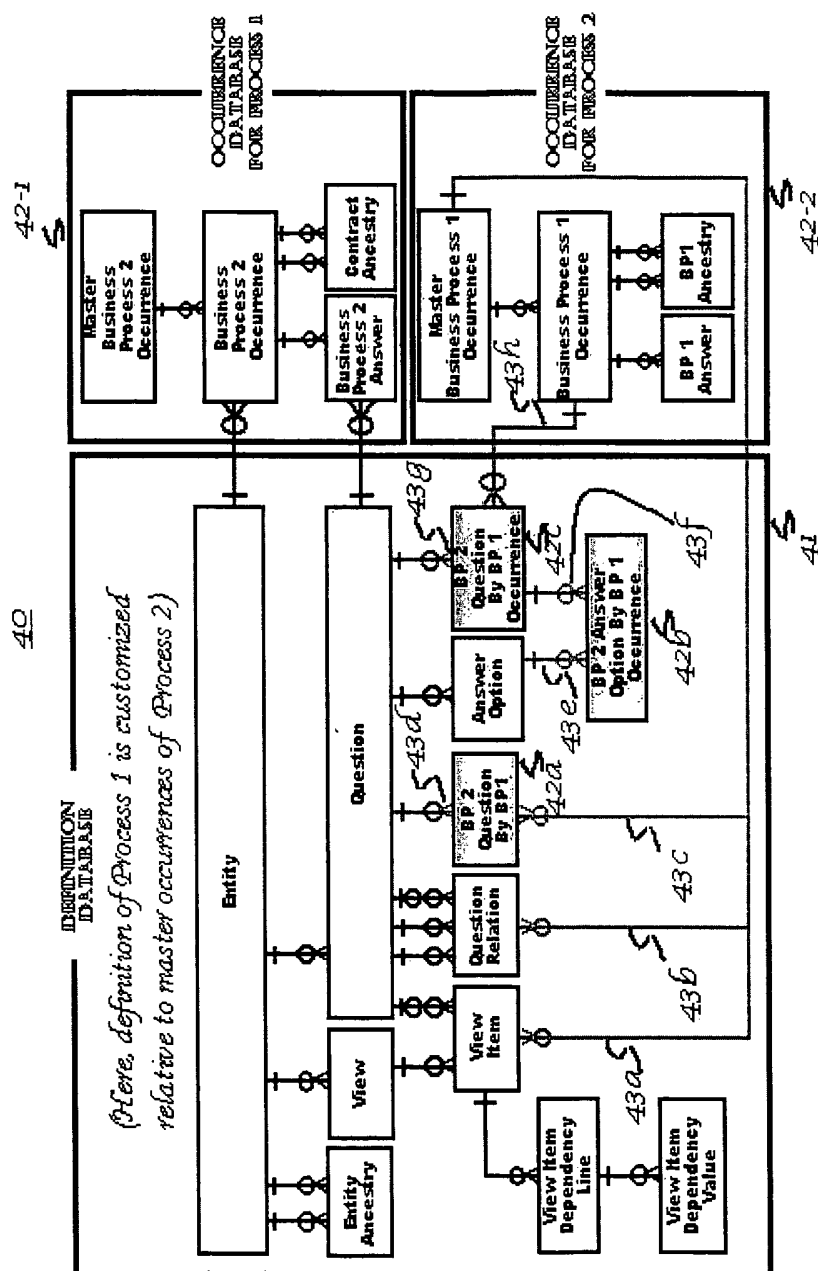
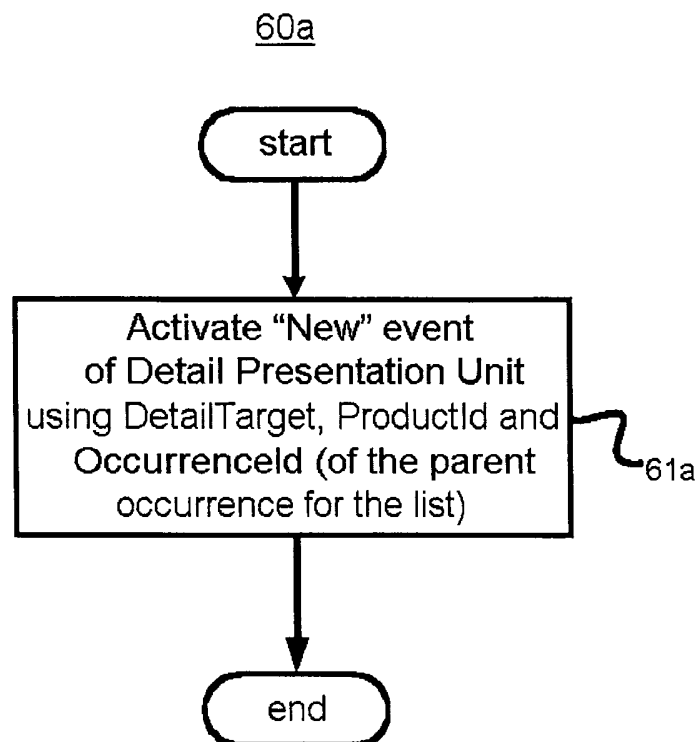


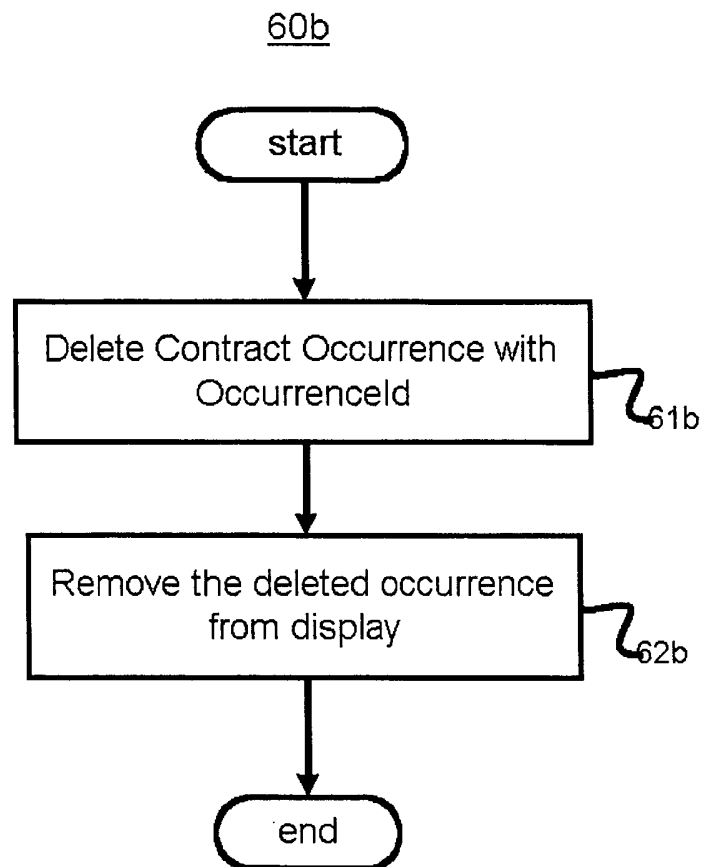
FIG 4

Events & Properties By Adaptive Unit

| | |
|--|--|
| Adaptive Unit: Events: Properties: | LIST Load, View/Edit, Add, Delete ViewName, DetailTarget, OccurrenceId, ProductId |
| Adaptive Unit: Events: Properties: | DETAIL Load, New, Continue, Submit ViewName, OccurrenceId, ProductId, SegmentNo |
| Adaptive Unit: Events: Properties: | UPDATEABLE LIST Load, Add Row, Delete Row, Submit ListViewName, DetailViewname, OccurrenceId, ProductId |
| Adaptive Unit: Events: Properties: | DUAL LIST Load, View/Edit, Add >>, <<Remove LeftViewName, RightViewName, DetailTarget, LinkedEntity, LinkedQuestion, OccurrenceId, ProductId |

FIG 5

**FIG 6A**

**FIG 6B**

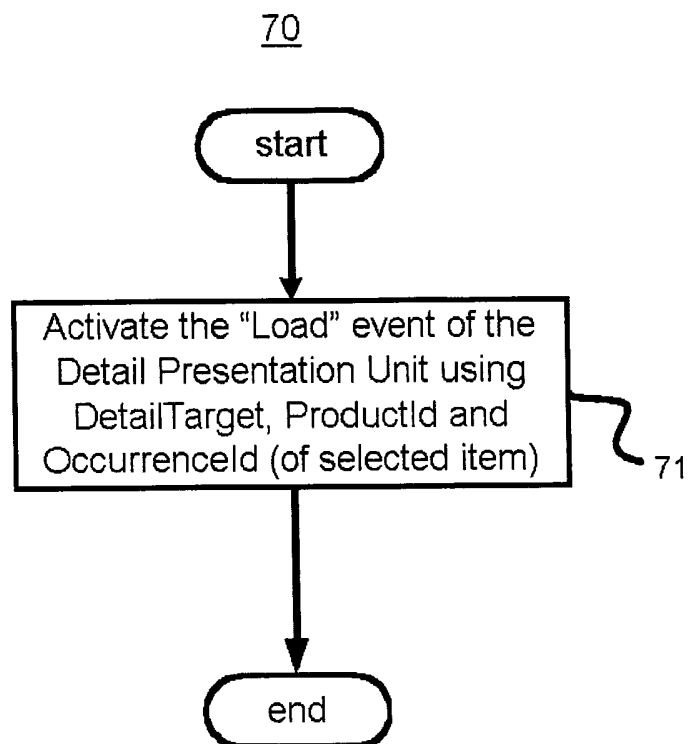
**FIG 7**

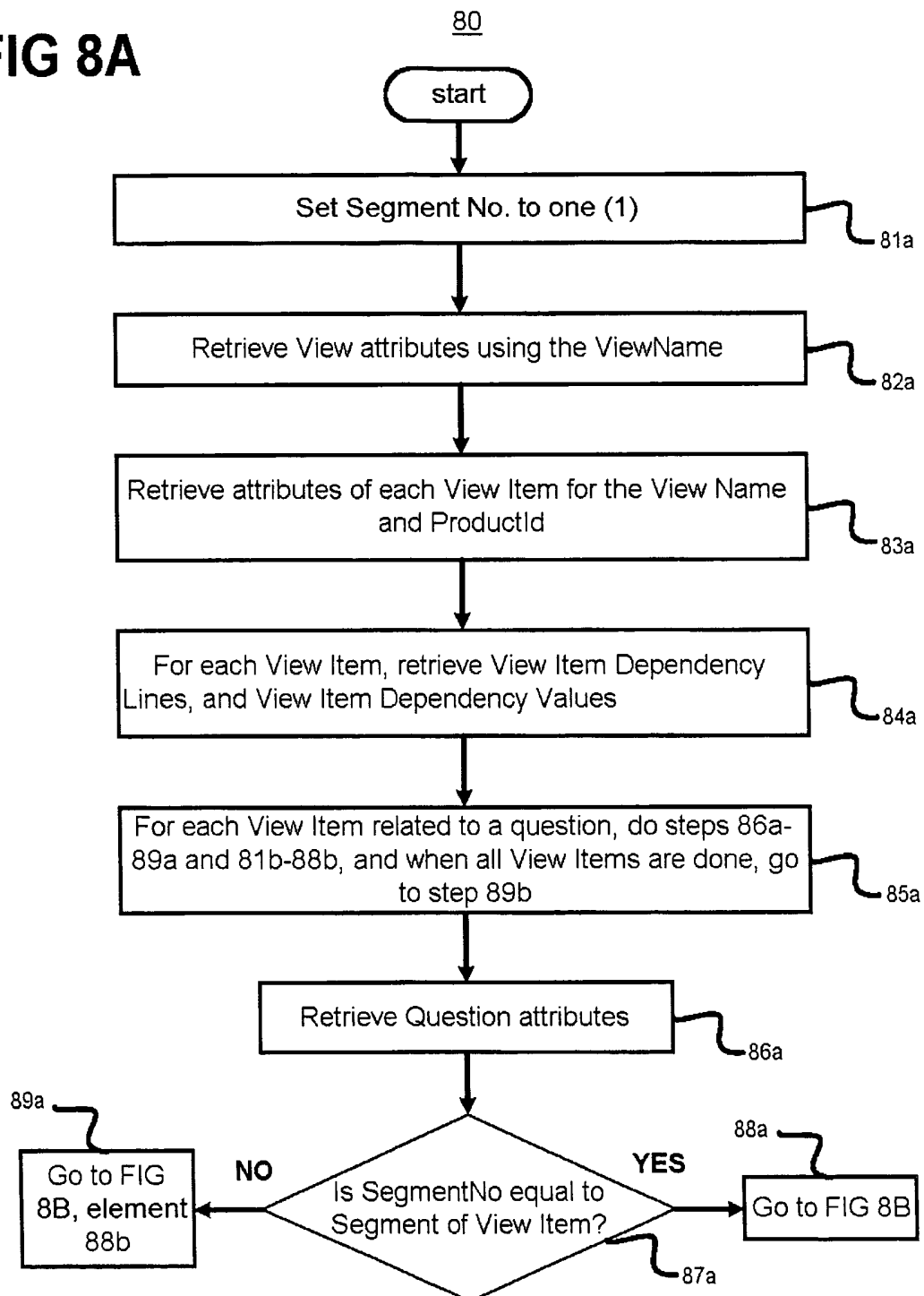
FIG 8A

FIG 8B

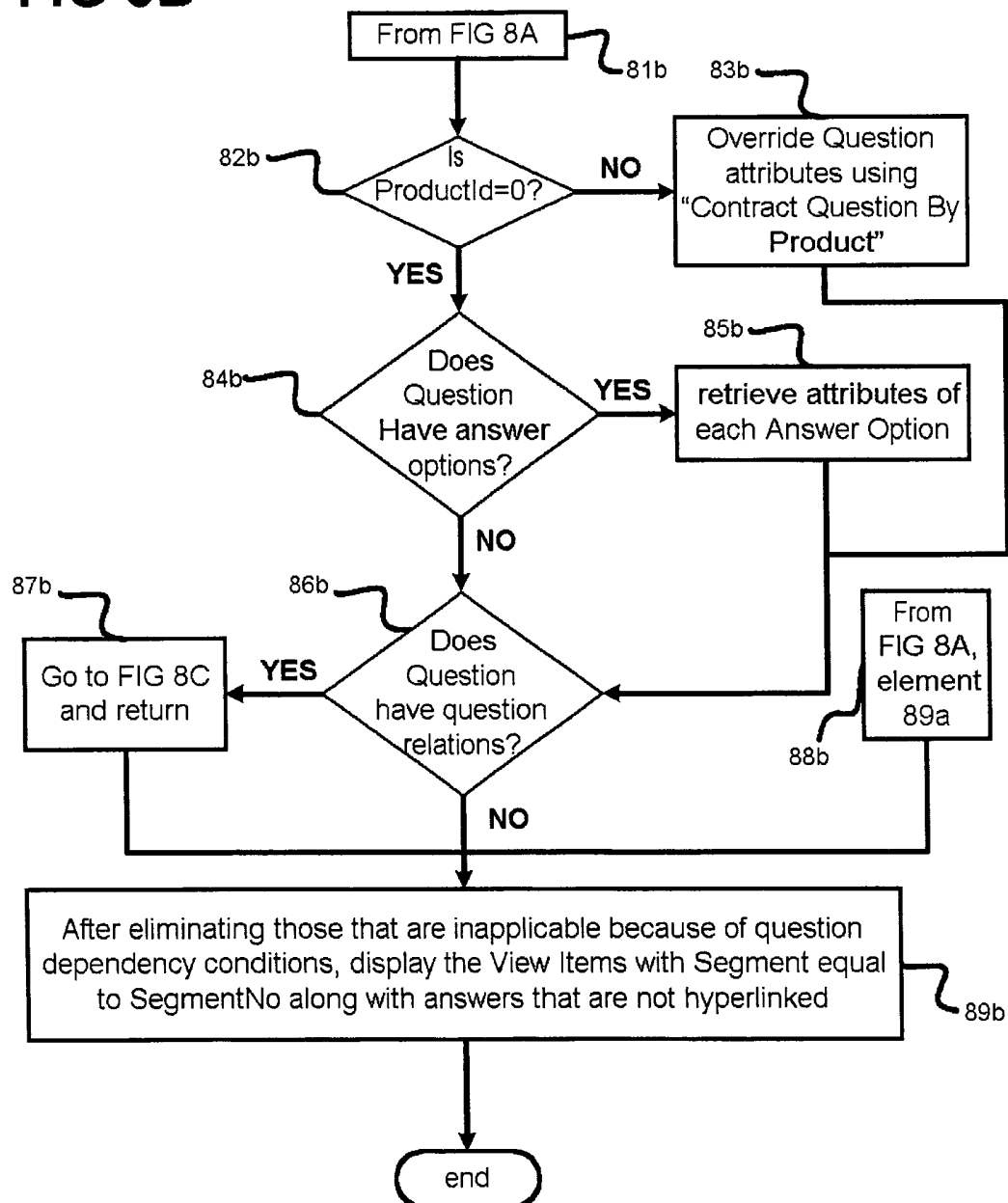


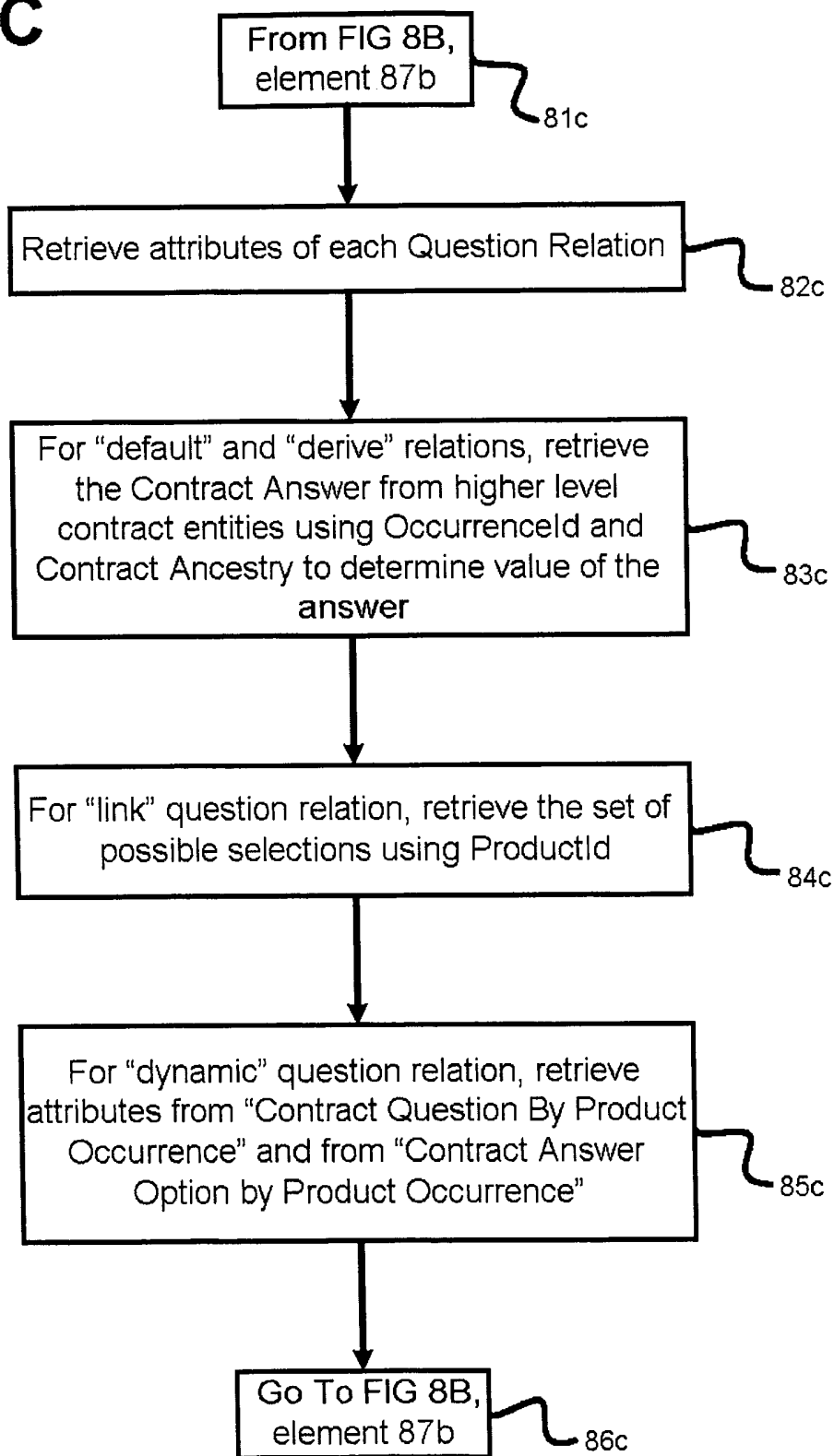
FIG 8C

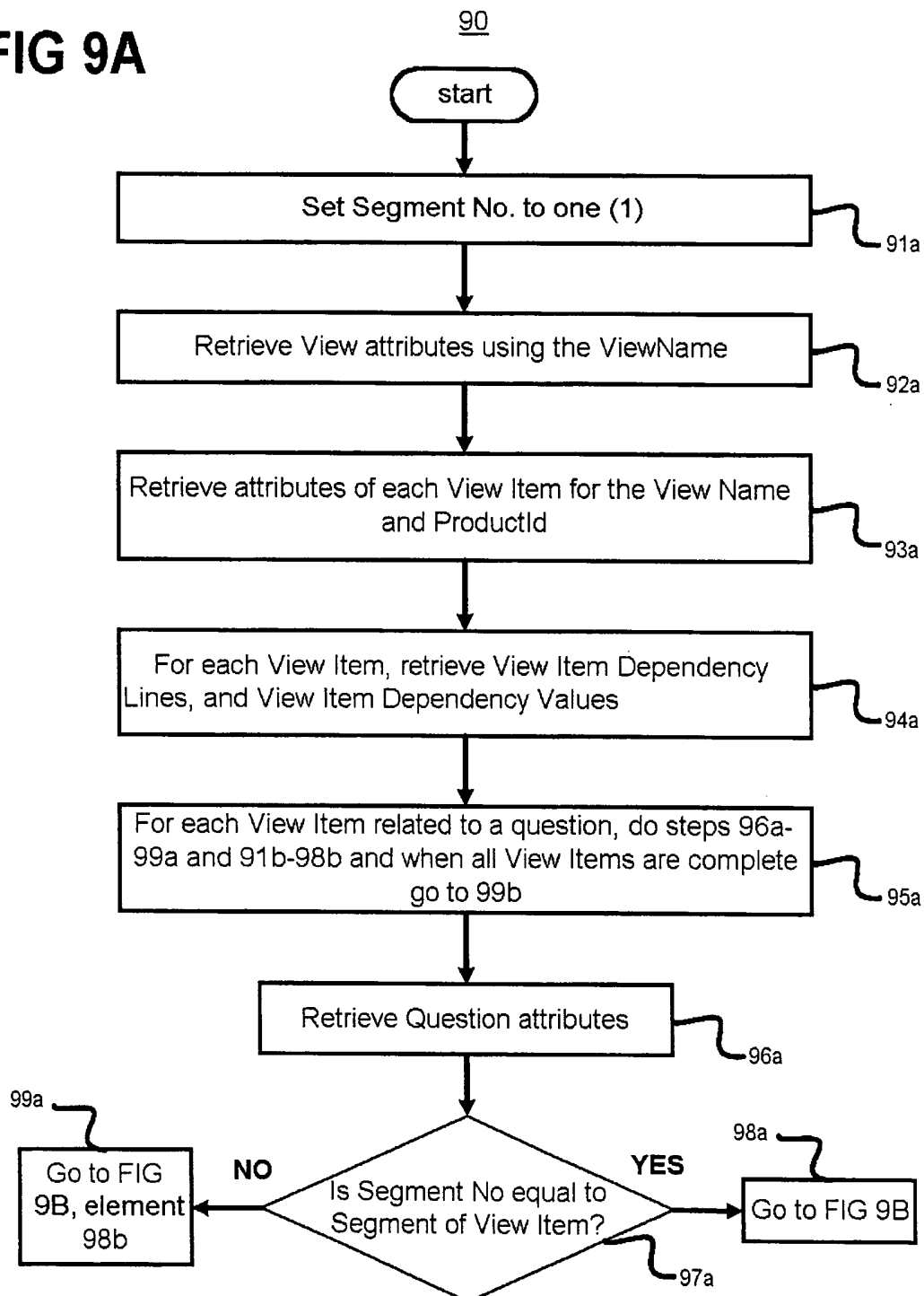
FIG 9A

FIG 9B

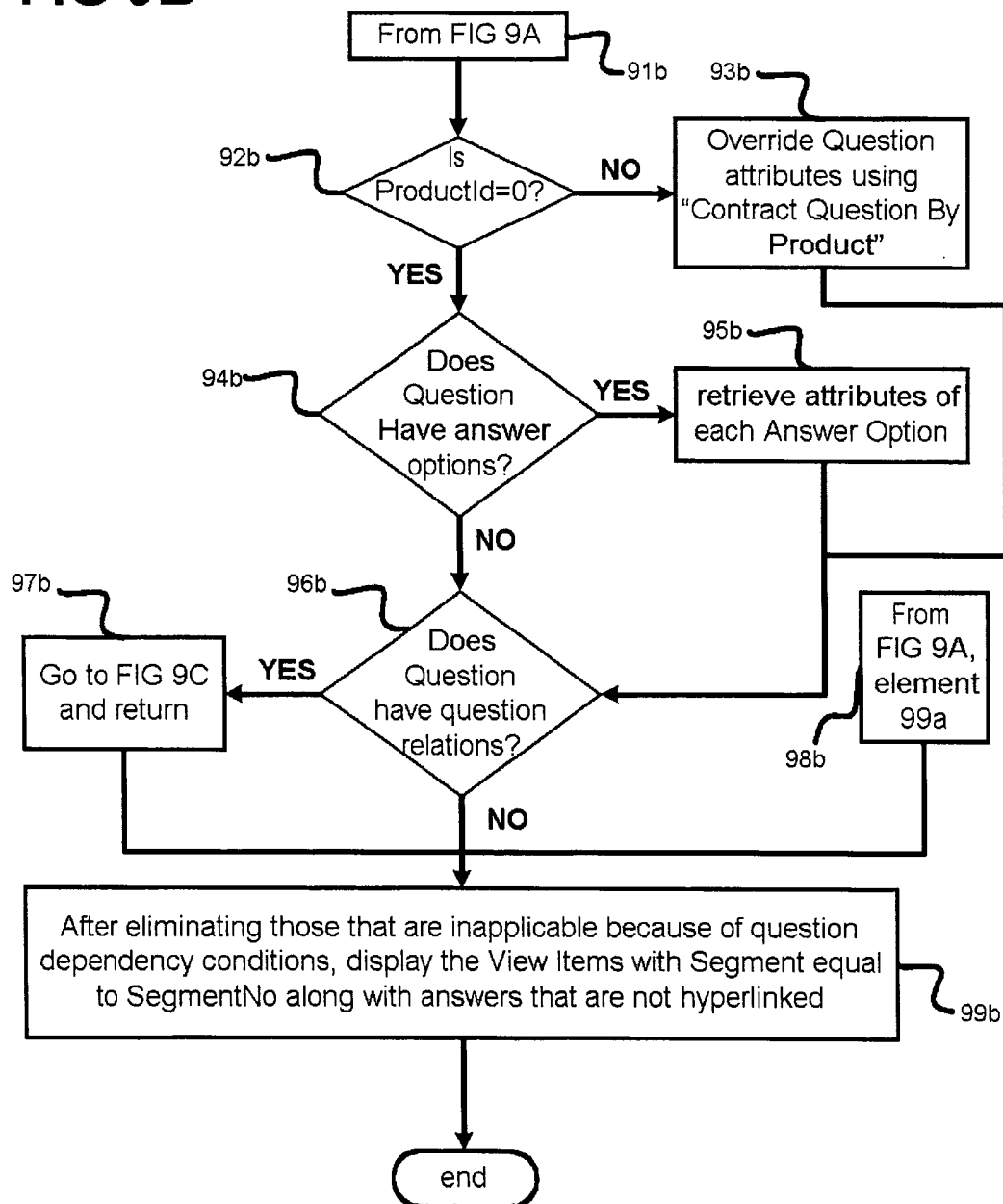


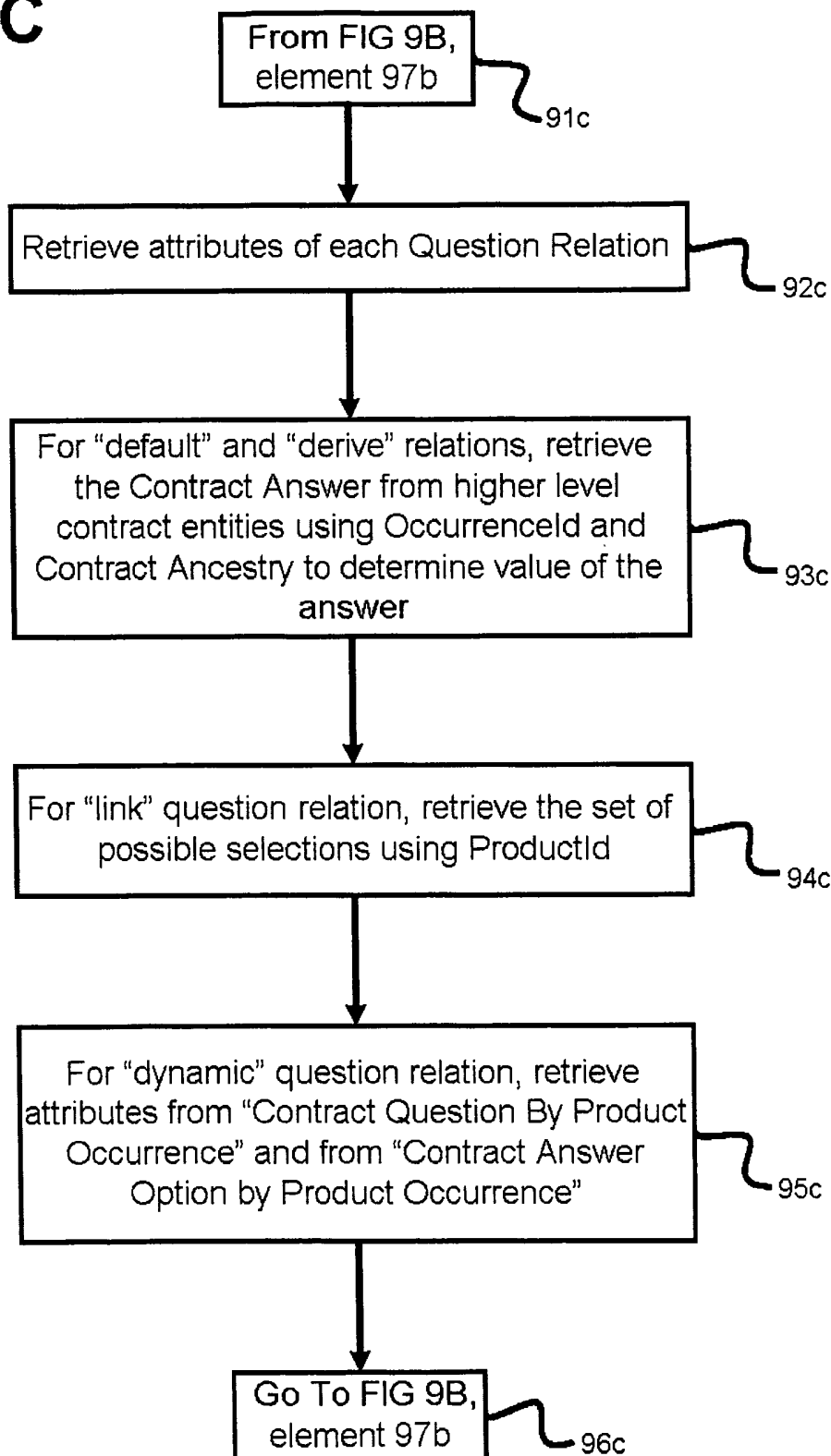
FIG 9C

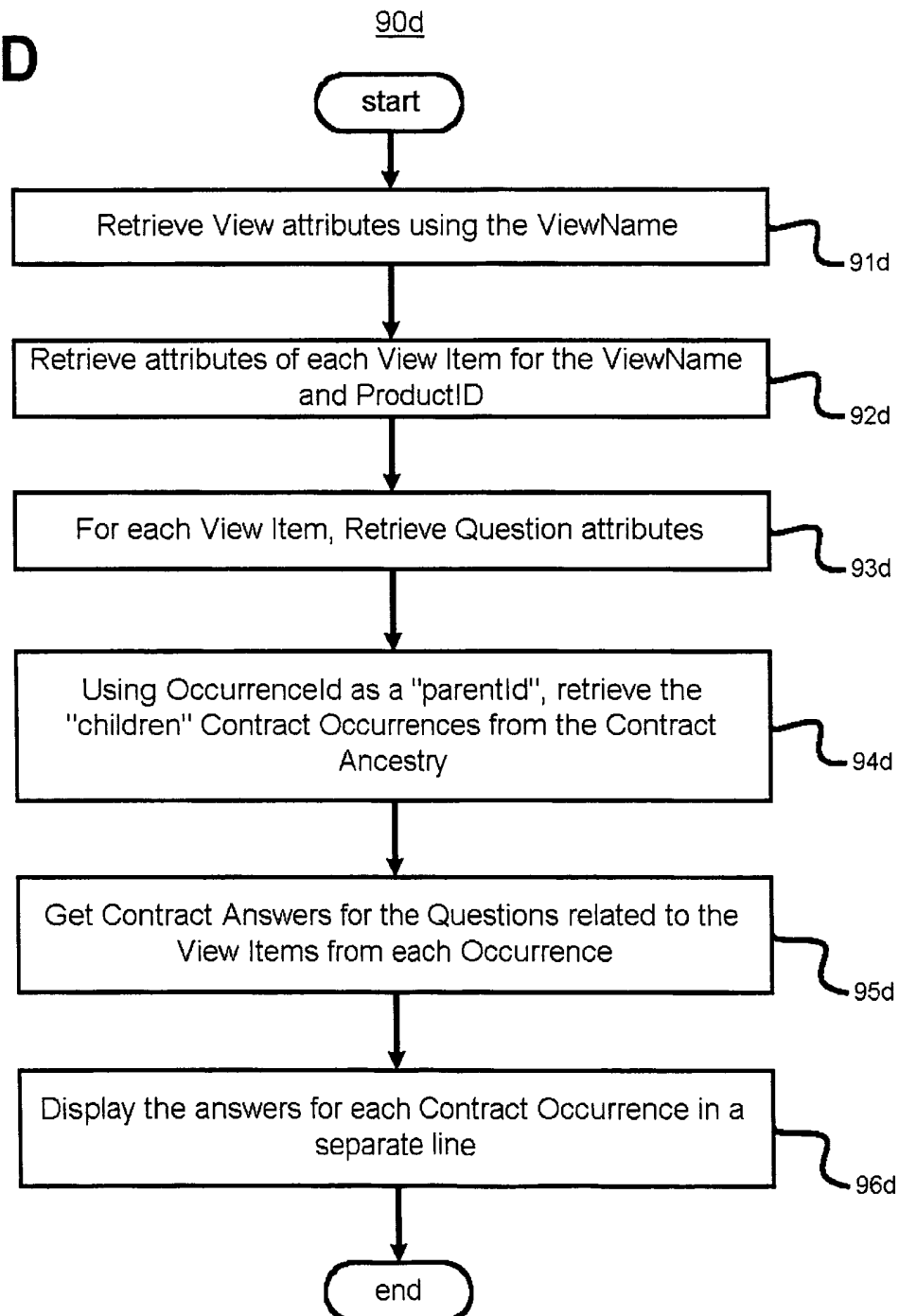
FIG 9D

FIG 10A

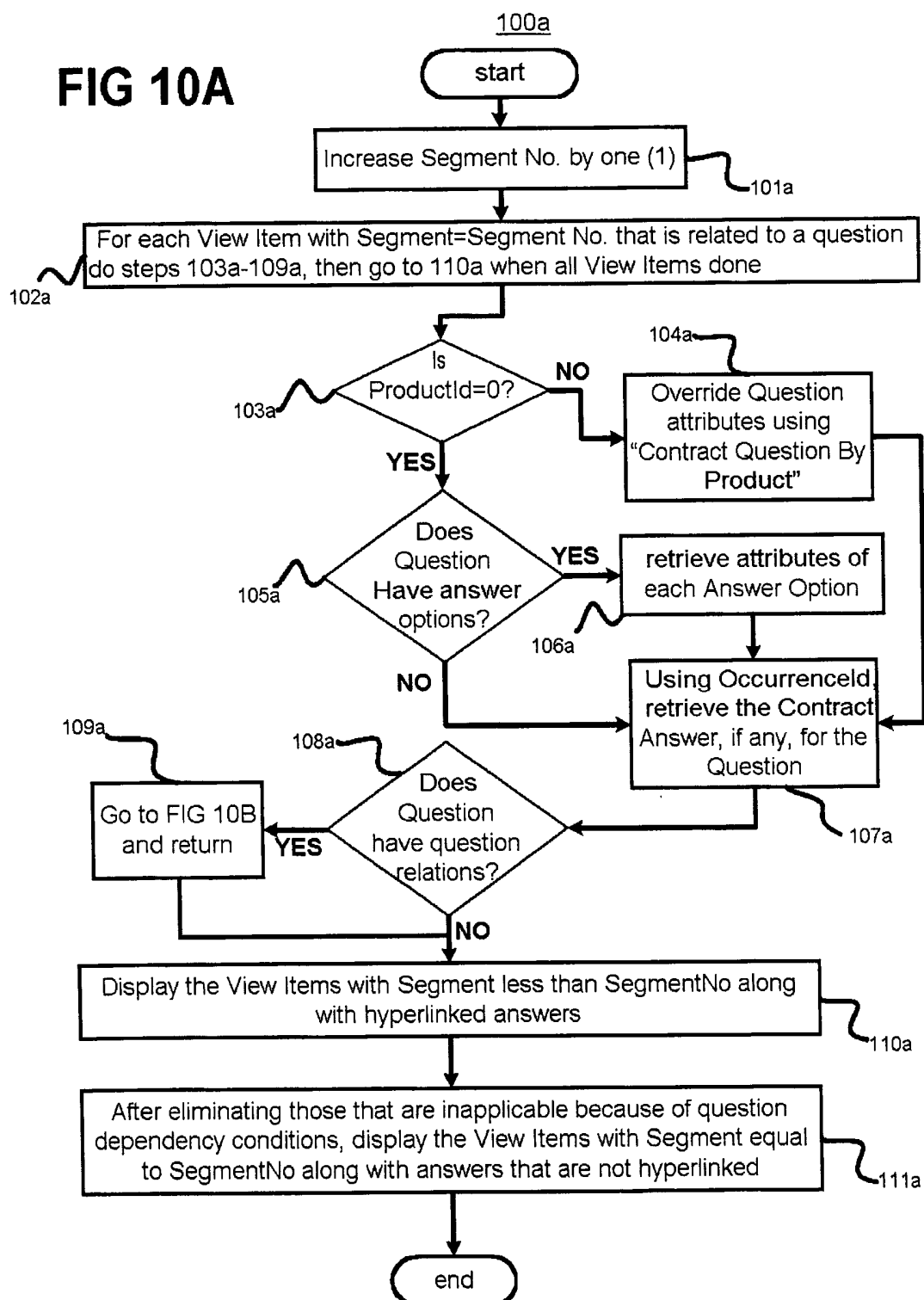


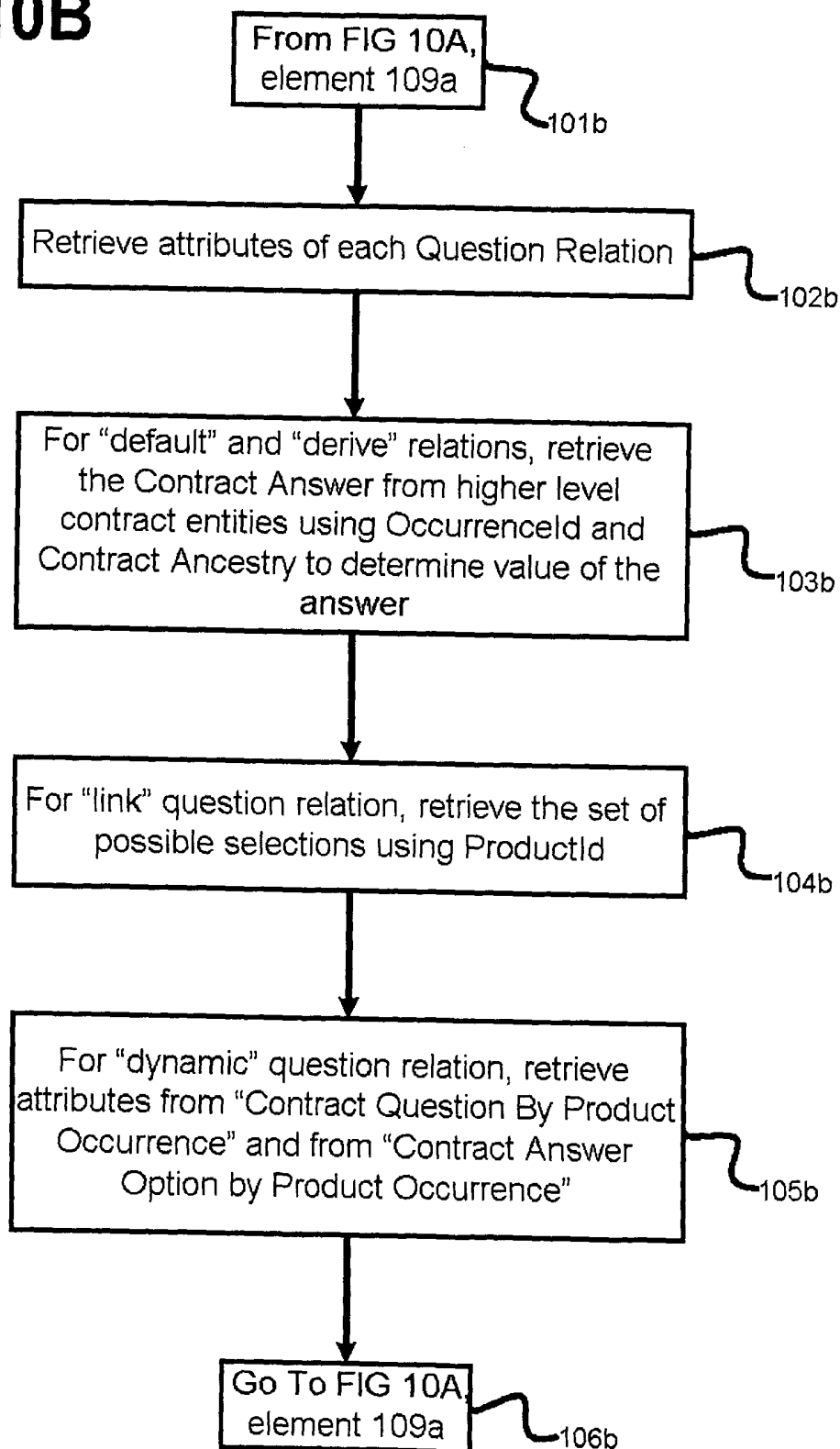
FIG 10B

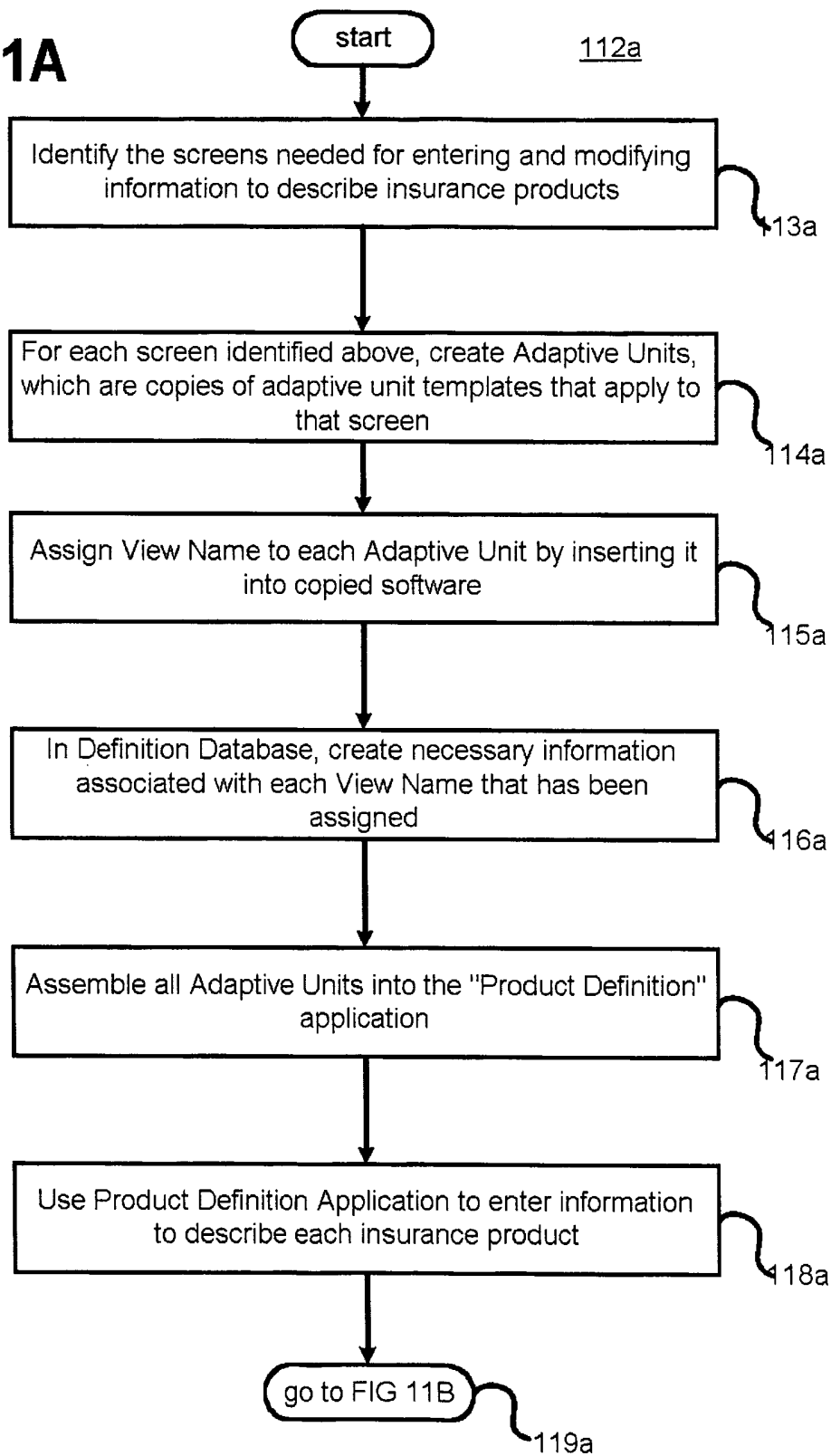
FIG 11A

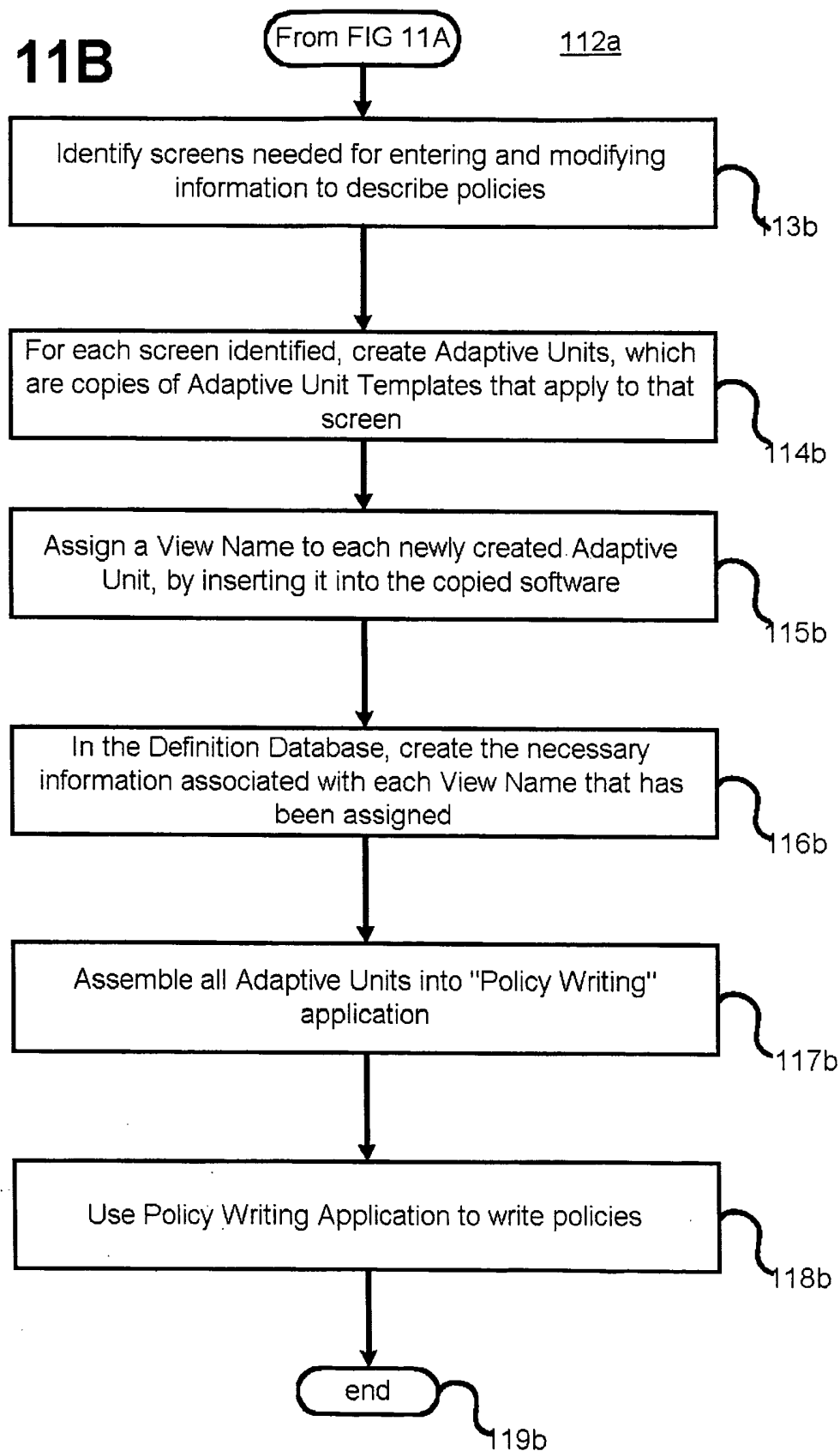
FIG 11B

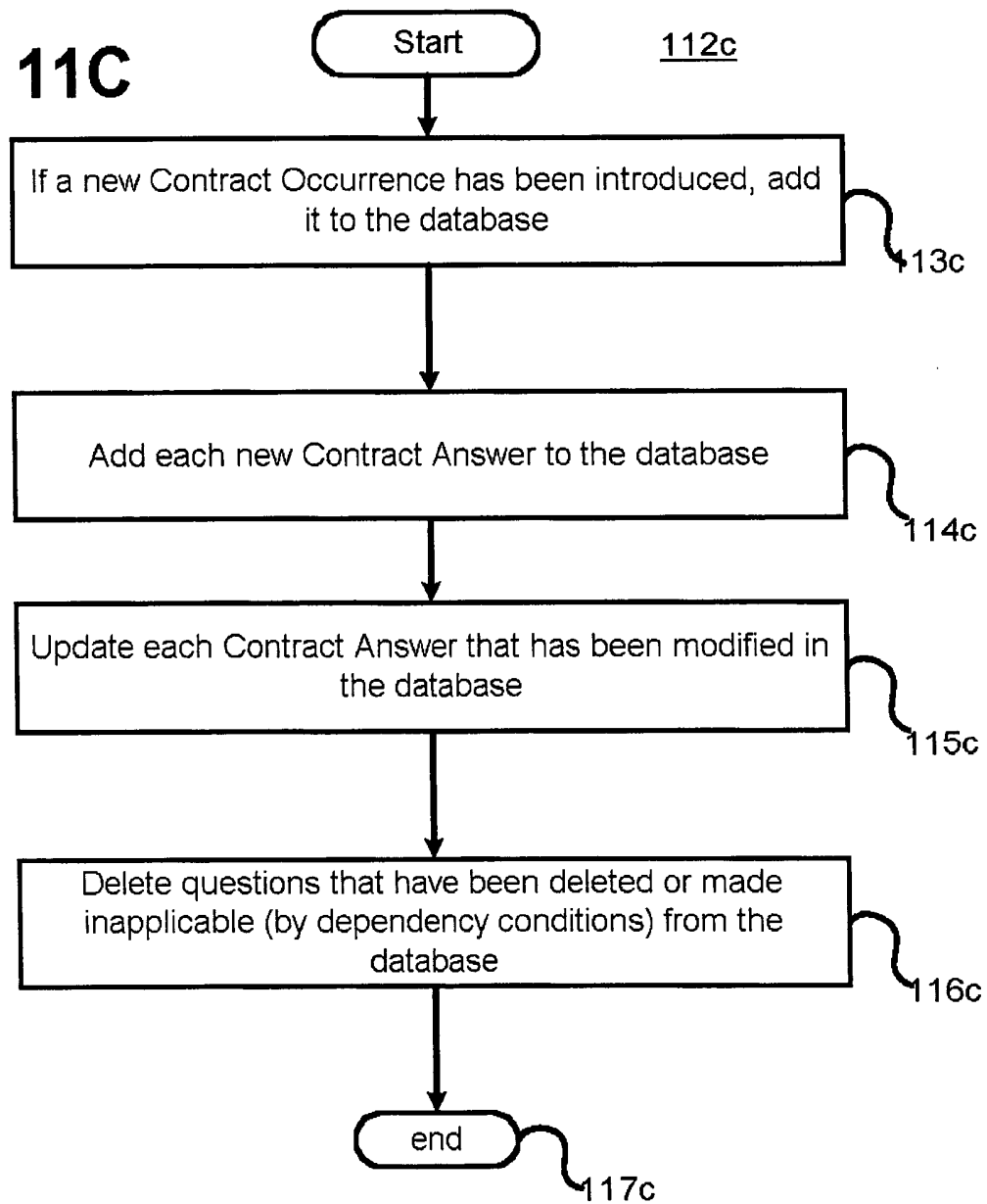
FIG 11C

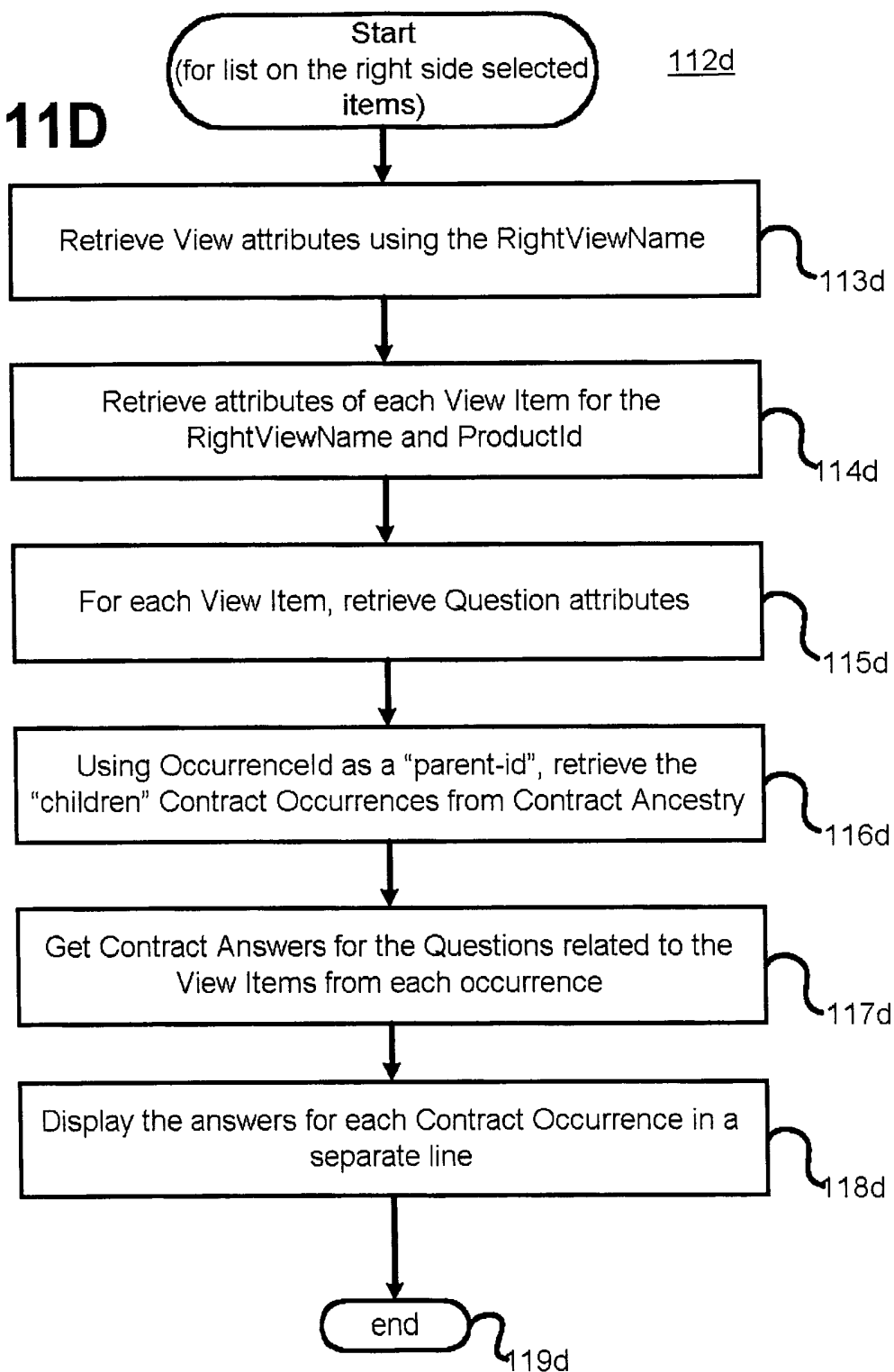
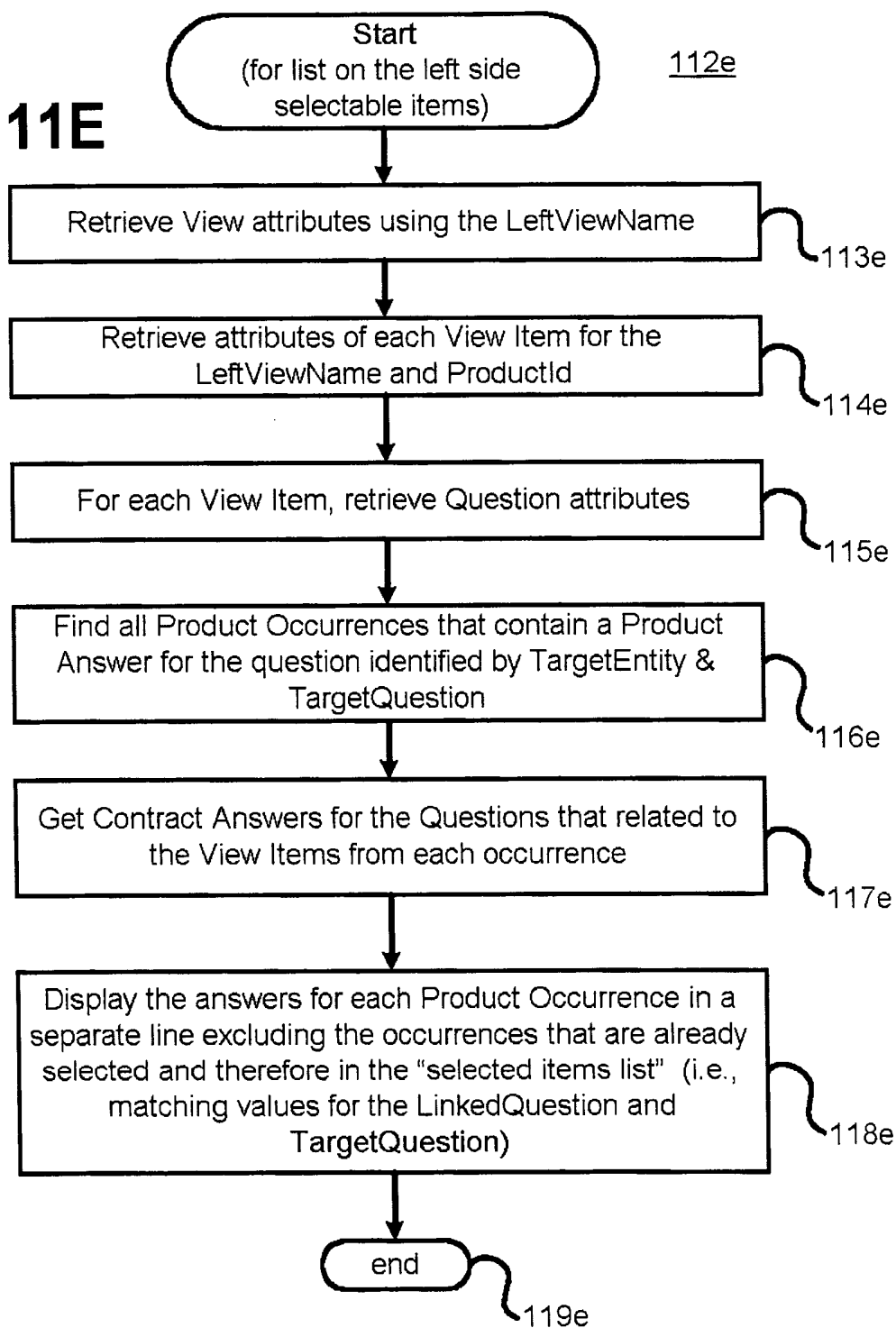
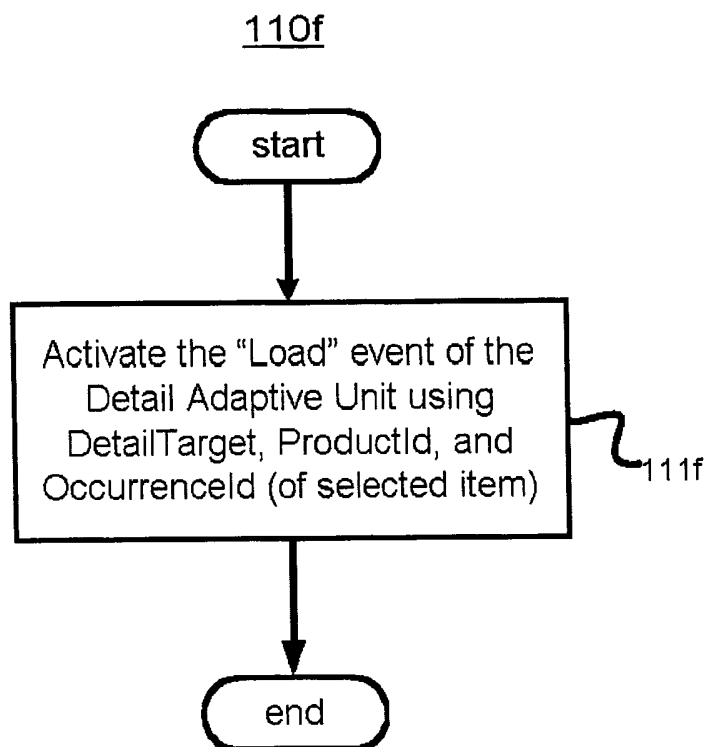
FIG 11D

FIG 11E

**FIG 11F**

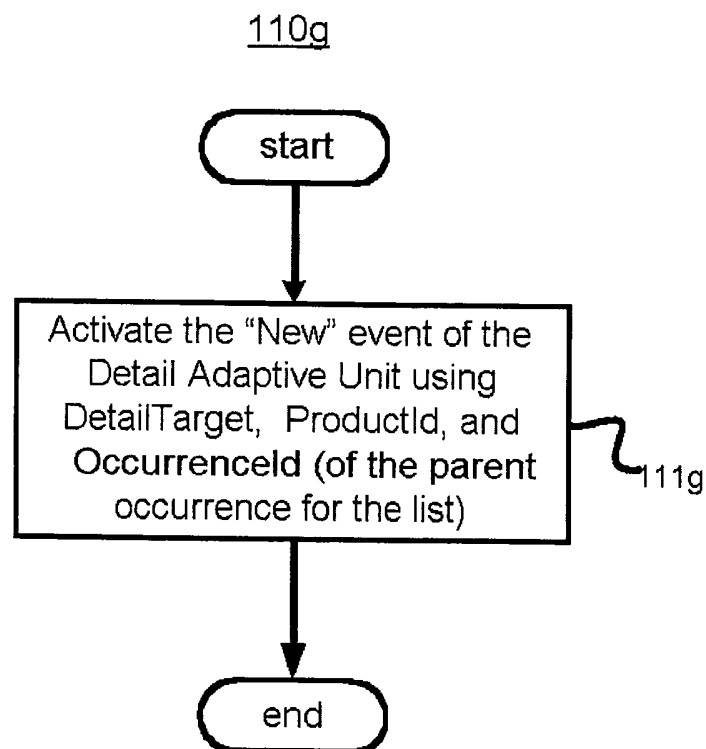
**FIG 11G**

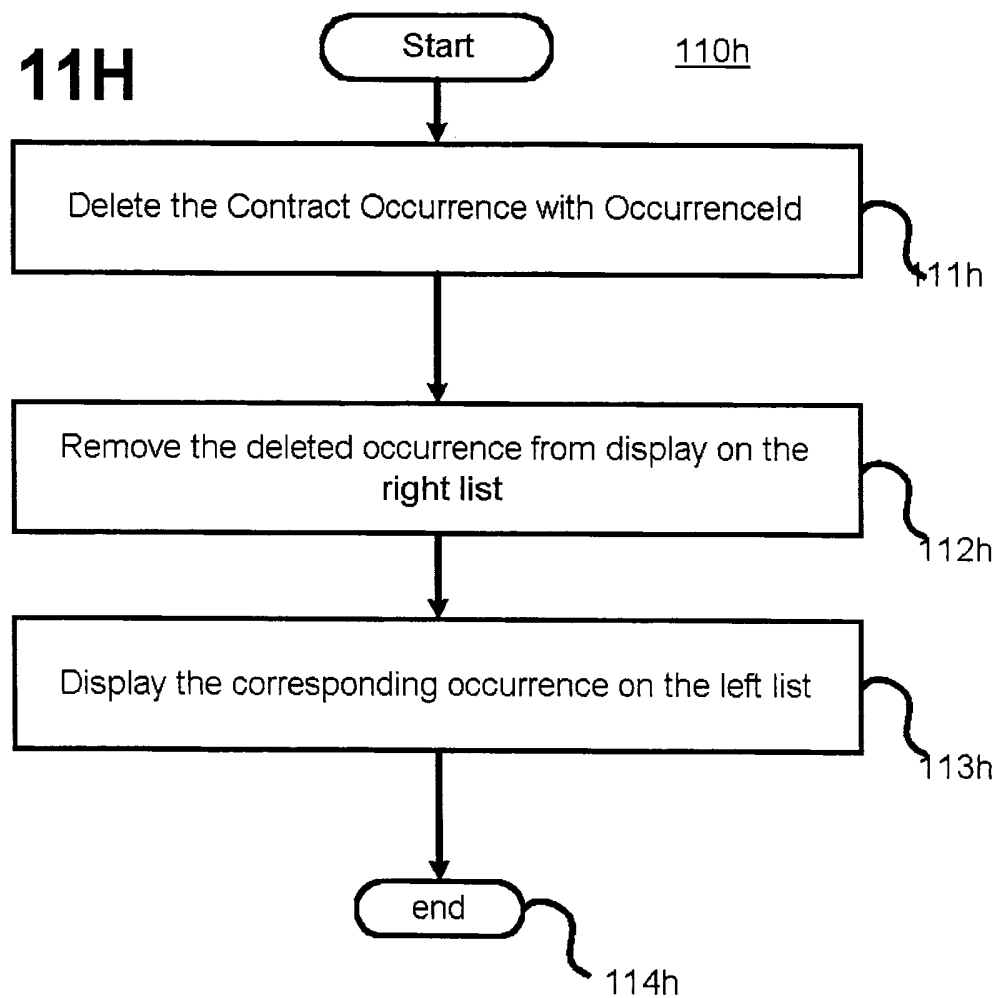
FIG 11H

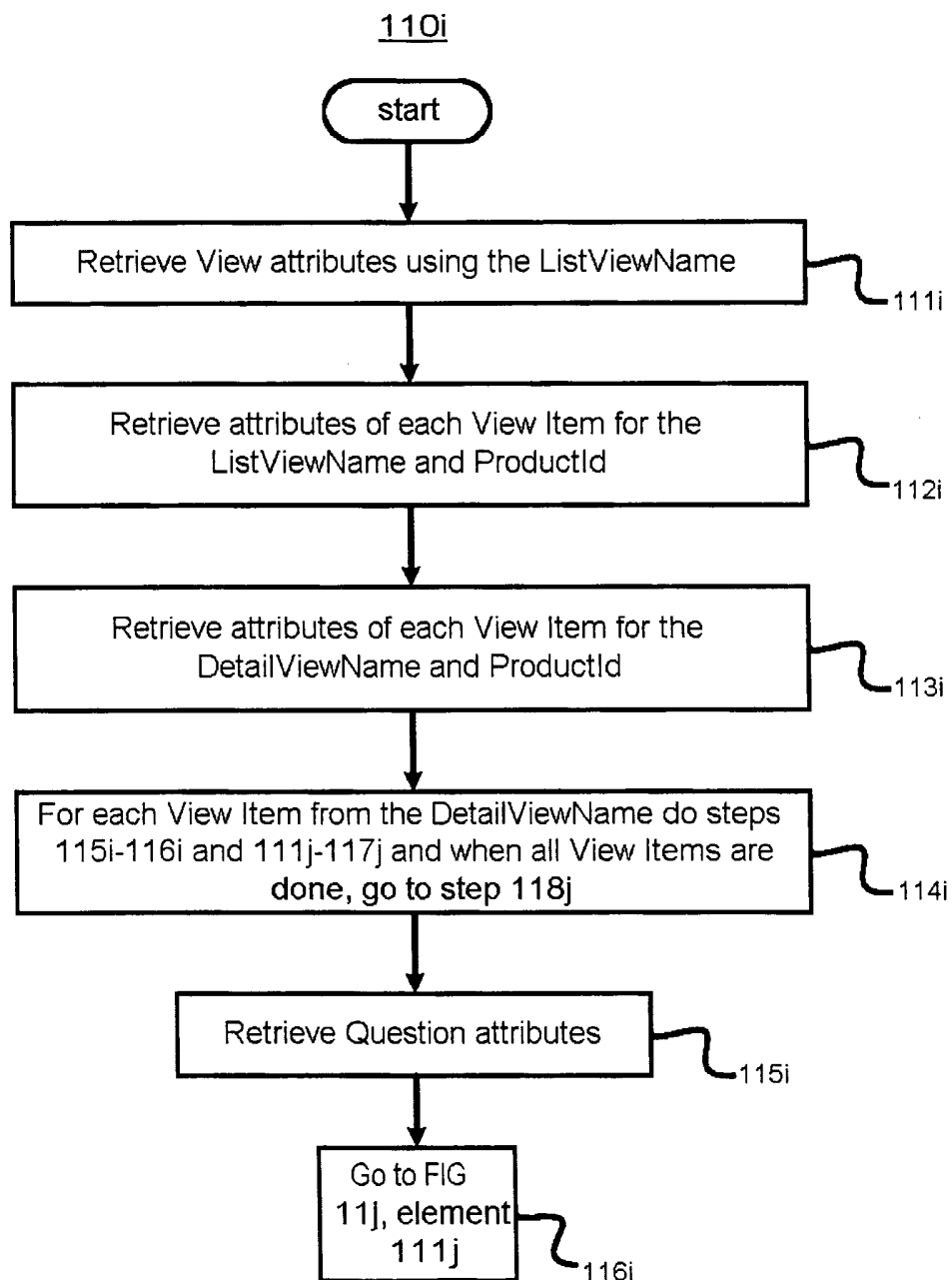
FIG 11I

FIG 11J

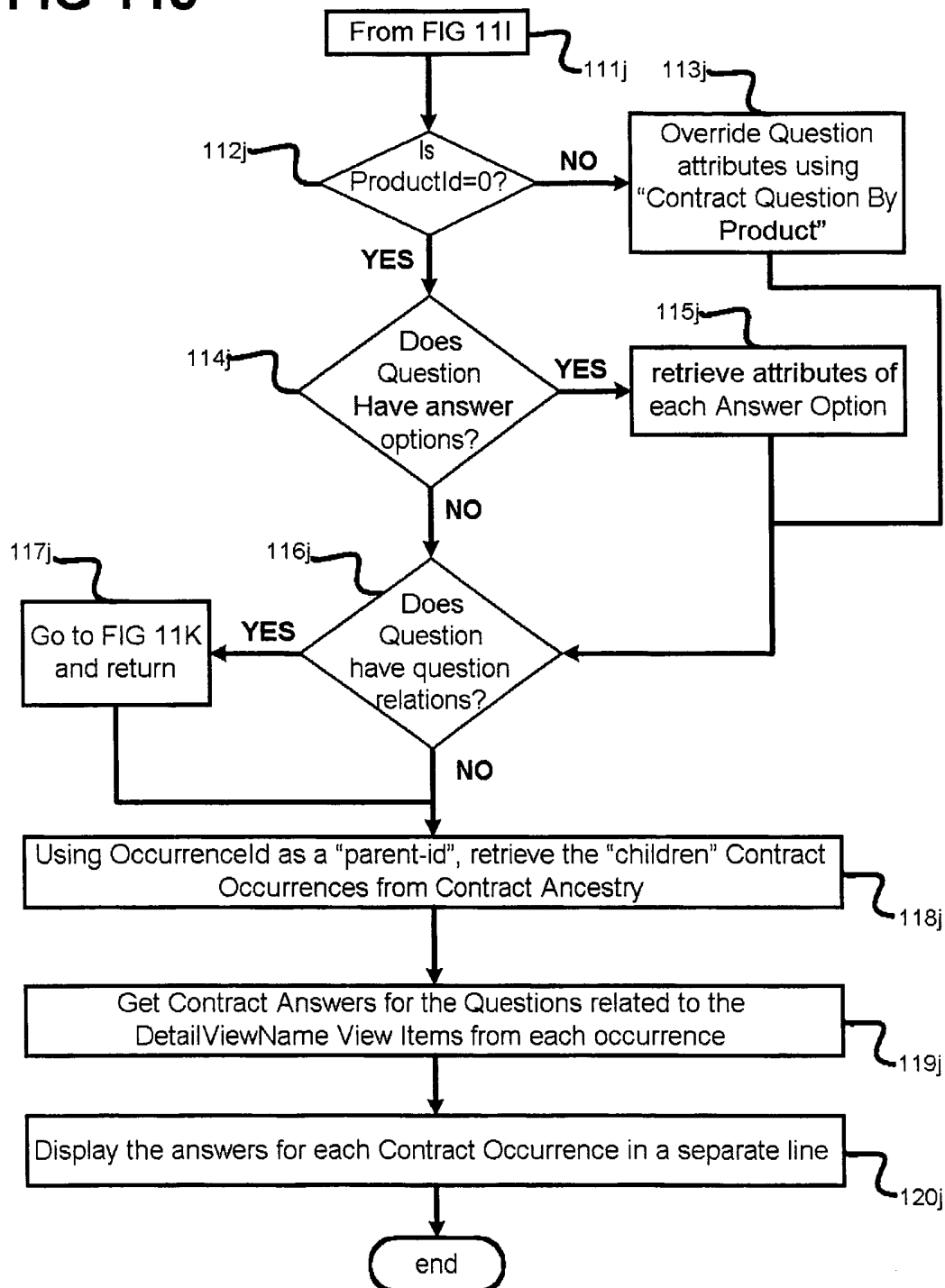
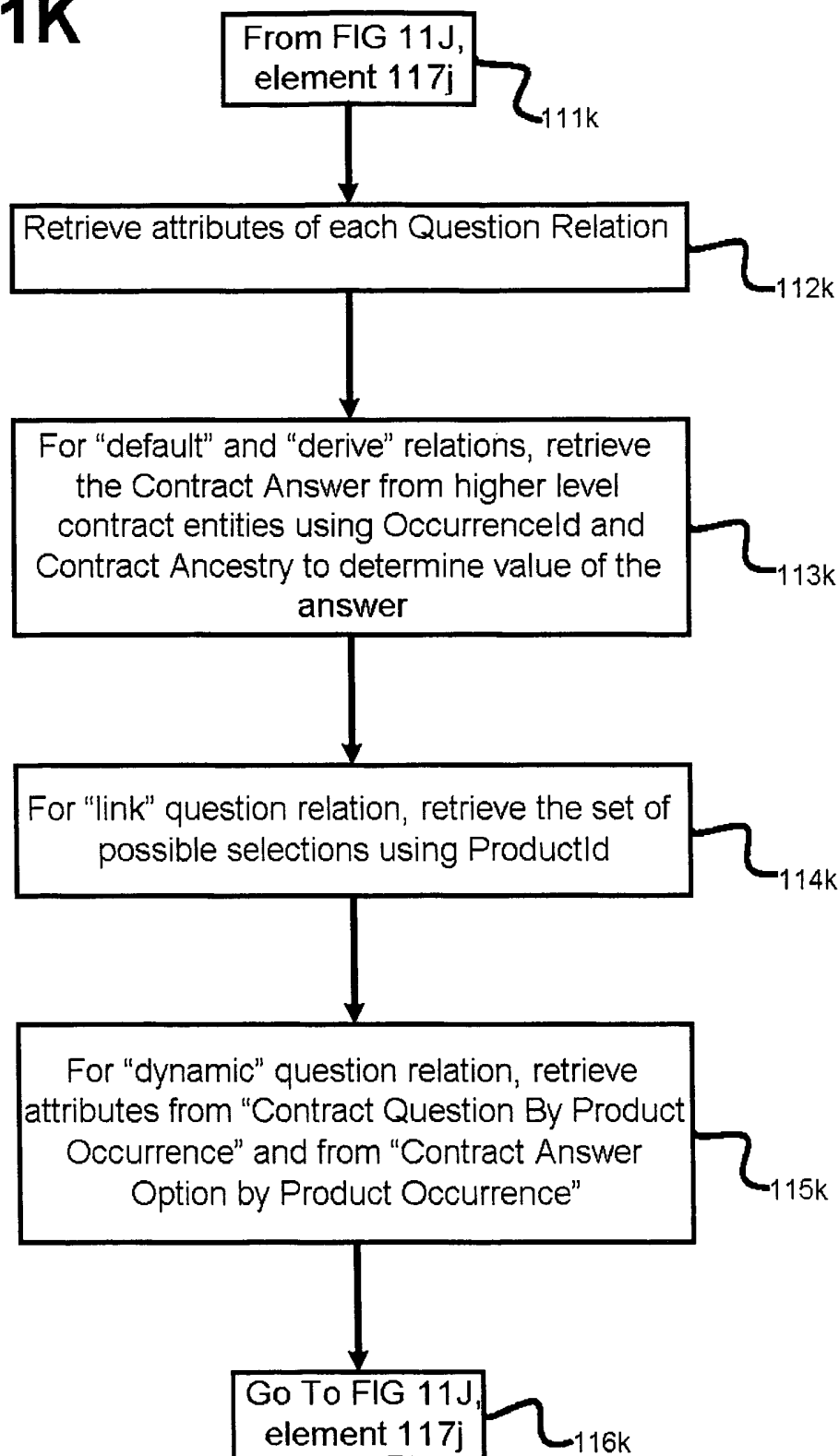
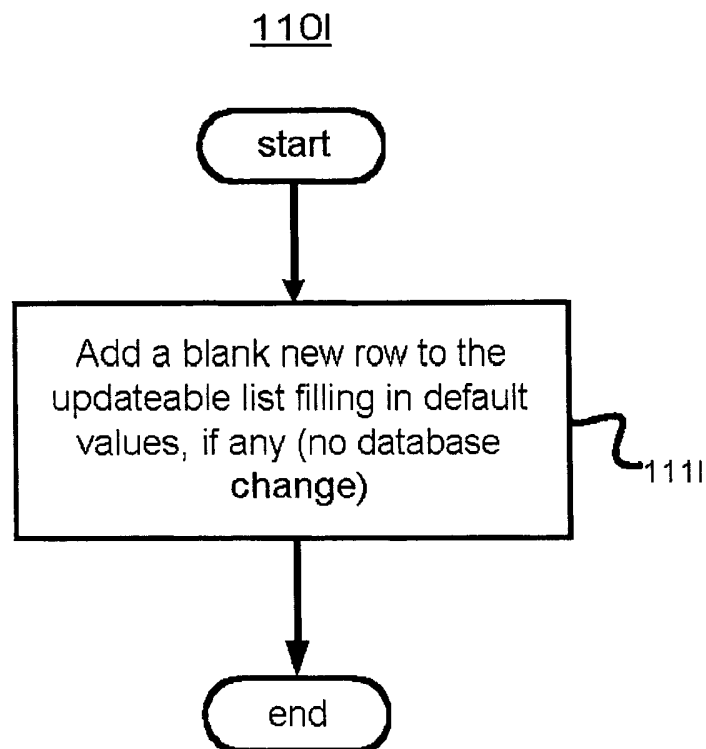


FIG 11K

**FIG 11L**

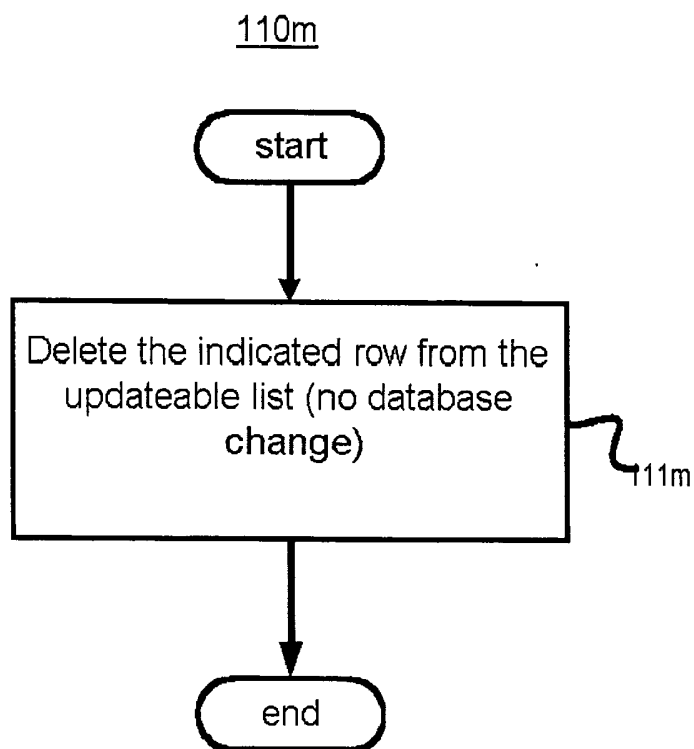
**FIG 11M**

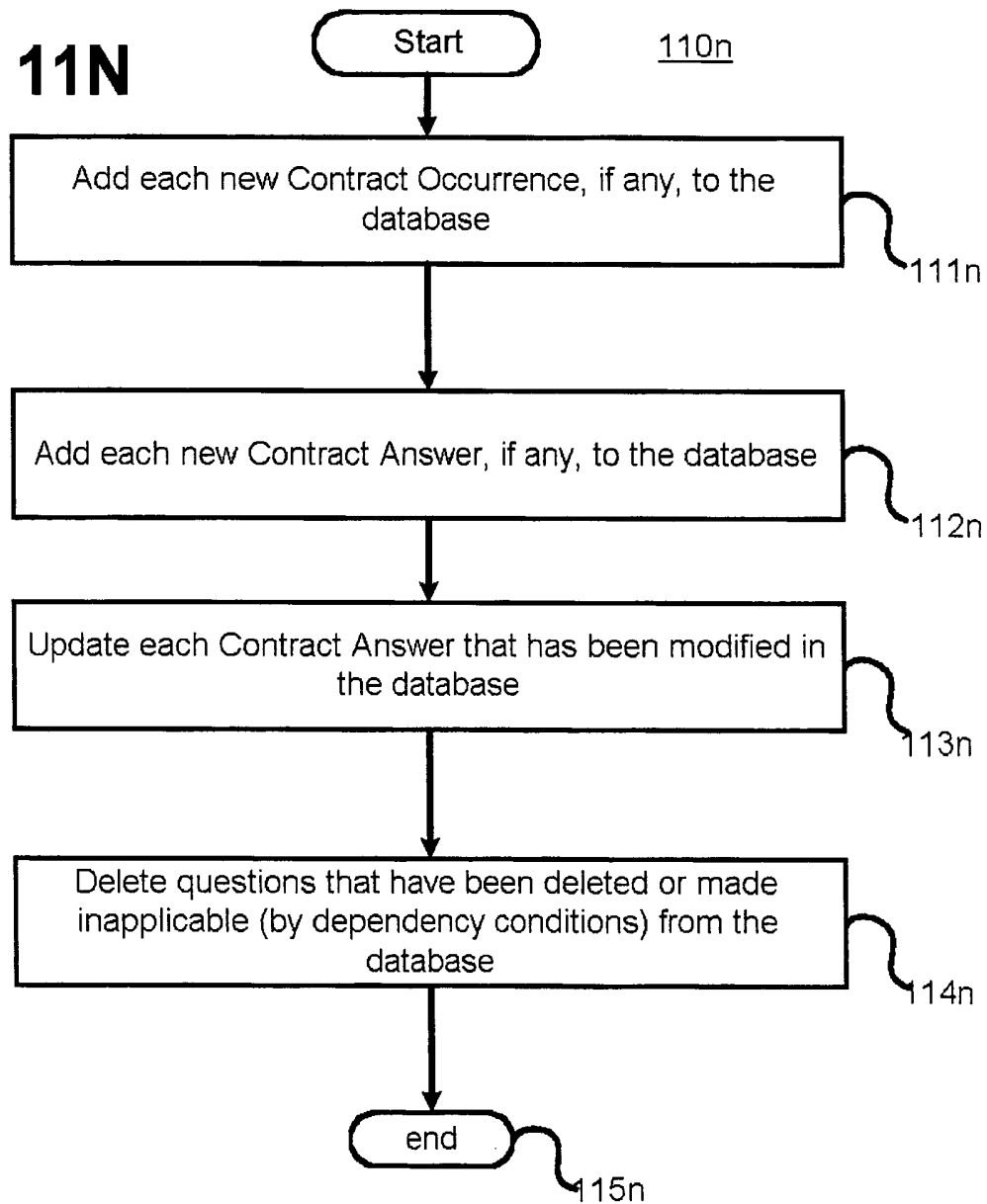
FIG 11N

FIG 12

| Policy List | | | |
|--------------------------------|----------------------------|--|--|
| Customer Name: ABC Corporation | Product: Ocean Marine | | |
| Producer Name: Walters & Ash | Rating State: Pennsylvania | | |
| Policy Number: 1234567890 | | | |

Header Area
(May include Site Links)

120

Help

< Back

Next >

Rate

FIG 13

| Policy List | |
|--------------------------------|----------------------------|
| Customer Name: ABC Corporation | Product: Ocean Marine |
| Producer Name: Walters & Ash | Rating State: Pennsylvania |
| Policy Number: 1234567890 | |

Context Area

131

[Policy](#)

[Vessel](#)

[Coverage](#)

[Participant](#)

[Help](#)
[Back](#)
[Next >](#)
[Print](#)

FIG 14

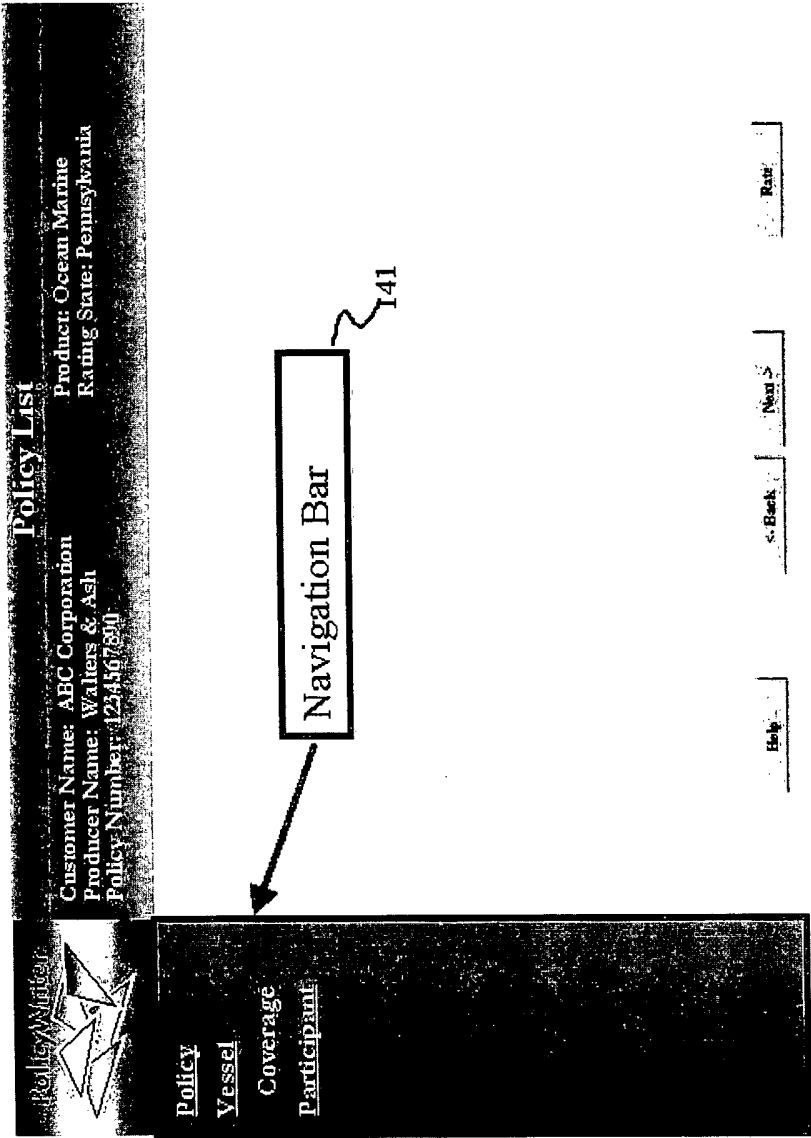



FIG 15

Policy List

Customer Name: ABC Corporation Product: Ocean Marine
 Producer Name: Walters & Ash Rating State: Pennsylvania
 Policy Number: 1234567890

| Policy | Vessel | Coverage | Participant |
|---|--------|----------|-------------|
|  | | | |

Signature Area 151

Help < Back Next > Rate

FIG 16

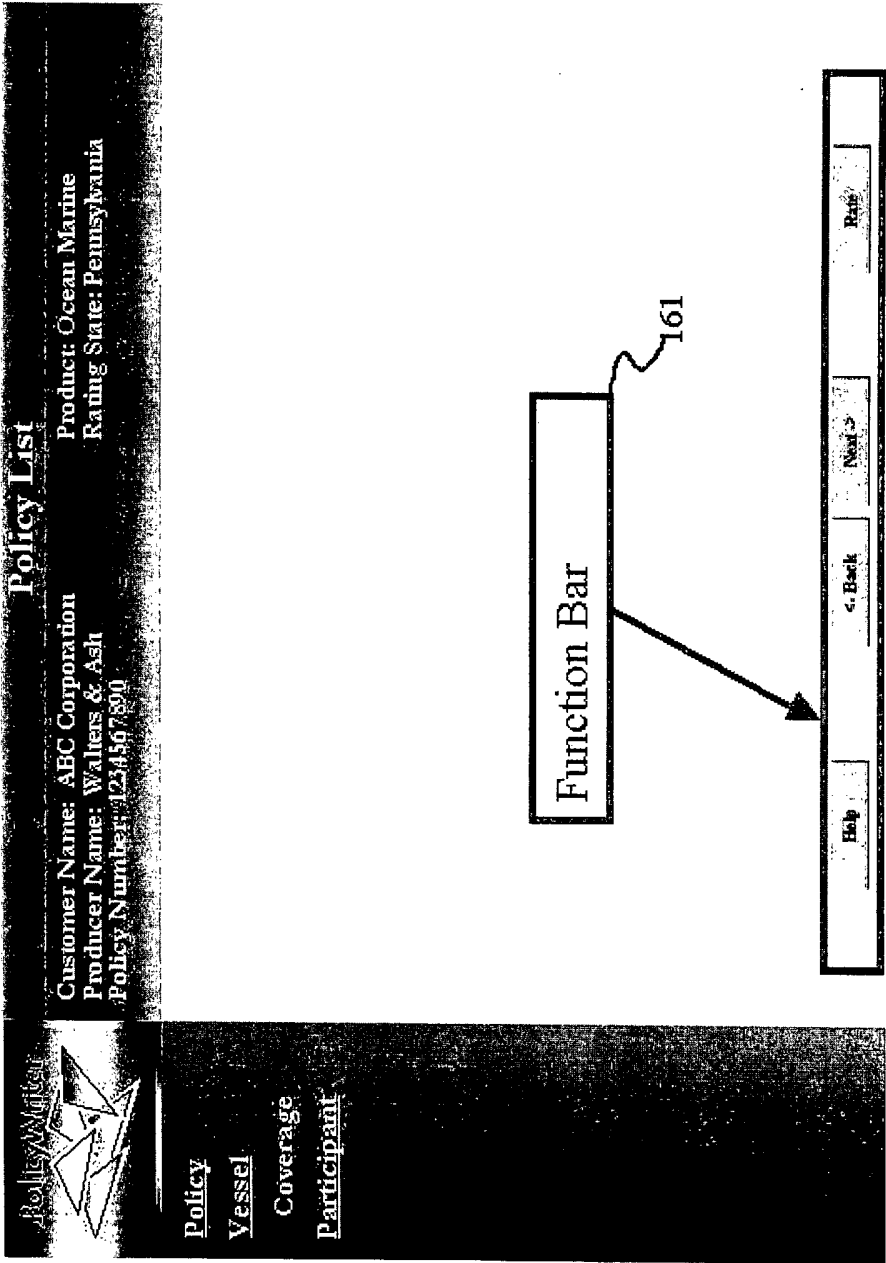



FIG 17

| | | | |
|--|--|--|--|
|  | | Policy List | |
| <p>Customer Name: ABC Corporation Producer Name: Walters & Ash Policy Number: 1234567890</p> | | <p>Product: Ocean Marine Rating State: Pennsylvania</p> | |

| | | | |
|---|--|--|--|
| <h2 style="text-align: center;">System Error</h2> <p style="text-align: center;">-14543 A severe database error has occurred. Please contact your system administrator.</p> | | | |
|---|--|--|--|

Error Display Area

171

| | | | |
|------|------|------|------|
| Help | Back | Next | Rate |
|------|------|------|------|

FIG 18

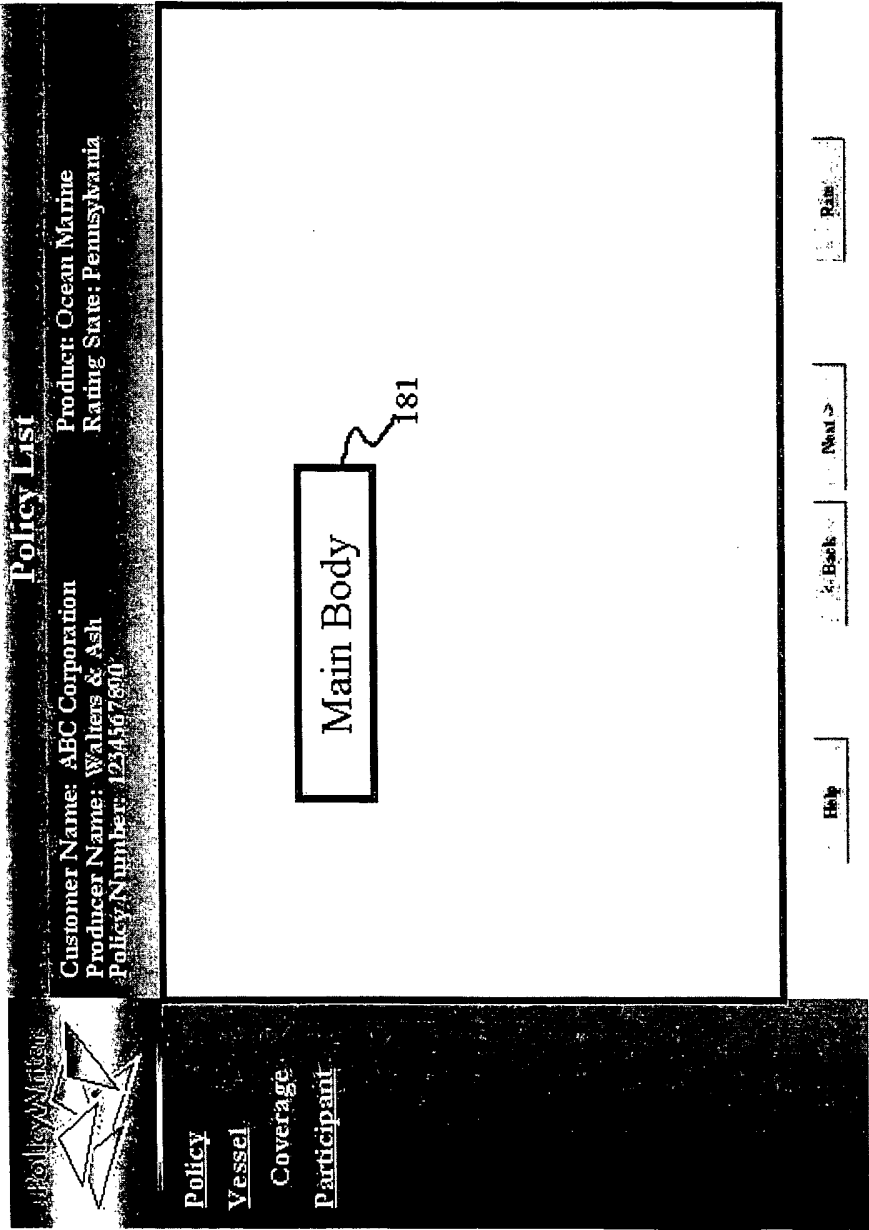



FIG 20

Detail Illustration

| POLICY NUMBER | | Vessel Detail | |
|--|--|--|--|
| Customer Name: ABC Corporation Producer Name: Walters & Ash Policy Number: 1234567890 | | Product: Ocean Marine Rating State: Pennsylvania | |
| Vessel Type: Yacht Base Sub Type: Motor | | Year: 2001 New (New Build): <input type="checkbox"/> | |
| Length (ft): 30 Manufacturer: C&C Yachts | | Model: M/40A Vessel Marking: | |
| Number of: 1 Engine Type: <input type="checkbox"/> Inboard <input type="checkbox"/> Outboard <input type="checkbox"/> Other | | Fuel Type: <input type="checkbox"/> Gas <input type="checkbox"/> Diesel | |
| Safety Equipment: <input type="checkbox"/> Vapor Detector <input type="checkbox"/> Depth Sounder <input type="checkbox"/> Radar <input type="checkbox"/> Automatic CO2 | | Winter Mooring Location: | |
| Summer Mooring Location: | | Address Line 1: 100 York St. Address Line 2: | |
| City: New York State: NY Zip: 10020-1234 | | City: Philadelphia State: PA Zip: 19101-1234 | |
| <input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Print"/> | | <input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Print"/> | |

FIG 21

Updateable List Illustration

| Policy | | Vessel | | Coverage | | Participant | |
|---|--------------------|--|---------------|--|---------------|---------------|---------------|
|  | | Vessel Operators Customer Name: ABC Corporation Producer Name: Waters & Ash Policy Number: 123456789 | | Product: Ocean Marine Rating State: Pennsylvania | | | |
| Operator Name | Operator Last Name | Date of Birth | Policy Number | Policy Number | Policy Number | Policy Number | Policy Number |
| 1 John | Smith | 01/01/75 | 10 | 123-45-6789 | 123-45-6789 | 123-45-6789 | PA 1 |
| 2 John | Smith | 01/01/75 | 20 | 123-45-6789 | 123-45-6789 | 123-45-6789 | FL 2 |
| 3 John | Smith | 01/01/75 | 30 | 123-45-6789 | 123-45-6789 | 123-45-6789 | CT 3 |

211

212

213

214

Back

Next

Cancel

FIG 22

Dual List Illustration

| POLICY | | Coverage Selection | |
|--------------------------|-----------------|--------------------|----------------------------|
| Policy Number | Product Name | Customer Name | Product Ocean Marine |
| 123456789 | ABC Corporation | Wahlers & Ash | Rating State: Pennsylvania |
| Policy Number: 123456789 | | | |

| Product Coverage Name | Coverage Code | Amount | Rate |
|-----------------------|---------------|--------|------|
| Actual Cash Value | FL 840 | 1500 | 1500 |
| Assign & Claim | FL 840 | 1500 | 1500 |
| Excess Coverage | FL 840 | 1500 | 1500 |
| Medical Payments | FL 840 | 1500 | 1500 |
| Personal Property | FL 840 | 1500 | 1500 |


| Policy | Vessel | Coverage | Participant |
|-----------|-----------------|---------------|----------------------------|
| 123456789 | ABC Corporation | Wahlers & Ash | Rating State: Pennsylvania |

222

221

FIG 23

List - Detail Illustration



Policy Vessel Coverage Participant

Customer Name: ABC Corporation

Producer Name: Waters & Ash

Policy Number: 1234567890

Product: Ocean Marine

Rating State: Pennsylvania

Coverage Selection

| Name | Street | City | State | Role |
|------------|-----------------|-----------|-------|-----------------|
| Colbeck | 1 Citibank Lane | New York | NY | First Mortgage |
| Farr Olson | 301 Broadway | Bethlehem | PA | Second Mortgage |

Participant Name

Address Line 1

Address Line 2

City

State

Zip

Email Address

Phone Number

Role

Specific Interest

Colbeck

1 Citibank Lane

New York

NY - New York

12345-1234

(717) 234-3232

First Mortgage

Add

Detail

Editing

Reset

231

Help

Submit

Reset

232

Back

Next

Date

FIG 24

Dual List - Detail Illustration

Policyholder

Policy

Vessel

Coverage

Participant

Coverage Selection

Customer Name: ABC Corporation

Producer Name: Walters & Ash

Policy Number: 1234567890

Product: Ocean Marine

Rating State: Pennsylvania

| Seq | Product Coverage | Coverage Code | Form Code | Coverage Name | Limit | Deductible |
|-----|-------------------|---------------|-----------|-------------------|-----------|------------|
| 6 | Annual Cash Value | PLACV | PL 840 | Annual Cash Value | \$200,000 | \$500 |
| 4 | Amateur & Classic | PLANT | PL 840 | Amateur & Classic | \$750,000 | \$500 |
| 5 | Crew Coverage | PLCREW | PL 840 | Crew Coverage | | |
| 7 | Medical Payments | PLMEDPAY | PL 840 | Medical Payments | | |
| 3 | Personal Property | PLPP | PL 840 | Personal Property | | |

241

Coverage Name: Personal Property

Limit: \$10,000

Deductible: \$500

242

Submit Reset

243


Back

Next

Exit

FIG 25

Detail – List Illustration



Policy
Vessel
Coverage
Participant

Customer Search
Customer Name: ABC Corporation
Producer Name: Walters & Ash
Policy Number: 1234567890
Producer Ocean Marine
Rating State: Pennsylvania

Customer Name:
State:
City:
Zip:
Customer ID:

Search Results

| Policy No. | Policy Name | Policy Type | Policy Status | Policy Date | Policy Location | Policy Agent | Policy Contact | Policy Email | Policy Phone | Policy Fax | Policy Address | Policy City | Policy State | Policy Zip | Policy Country |
|------------|-------------|-------------|---------------|-------------|-----------------|--------------|----------------|--------------|--------------|------------|----------------|-------------|--------------|------------|----------------|
| 1234567 | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal |
| 9876543 | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal | Qian Metal |

251

252

FIG 26
Question Source: Linked (L)

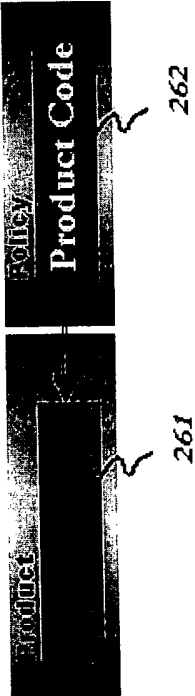


FIG 27

Question Source: Linked (cont.)

Example

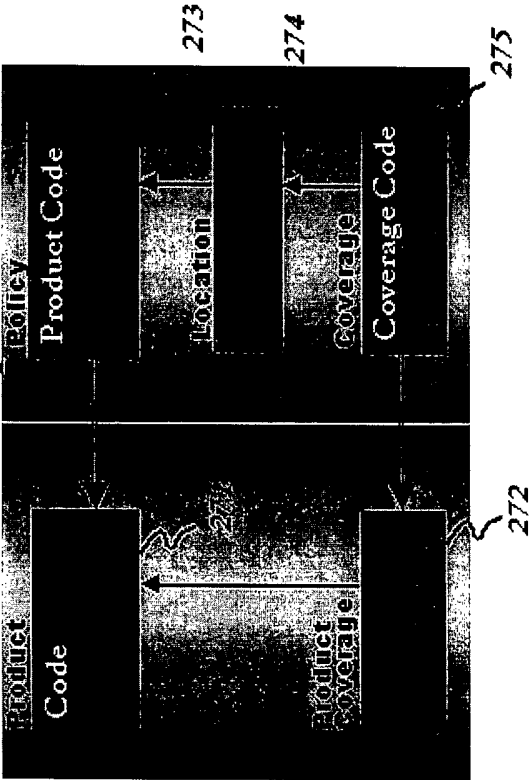


FIG 30

Question Source: Derived (cont.)

Example 2

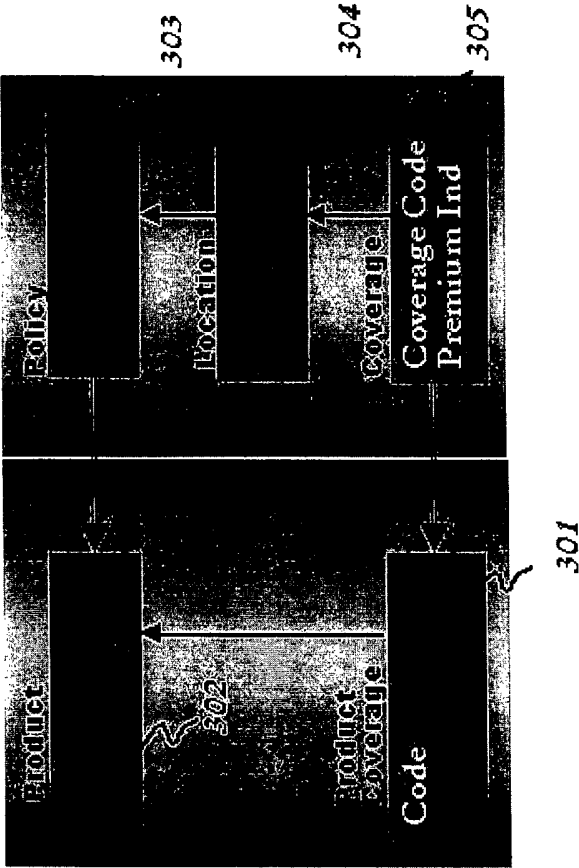


FIG 31
Question Source: Defaulted (F)

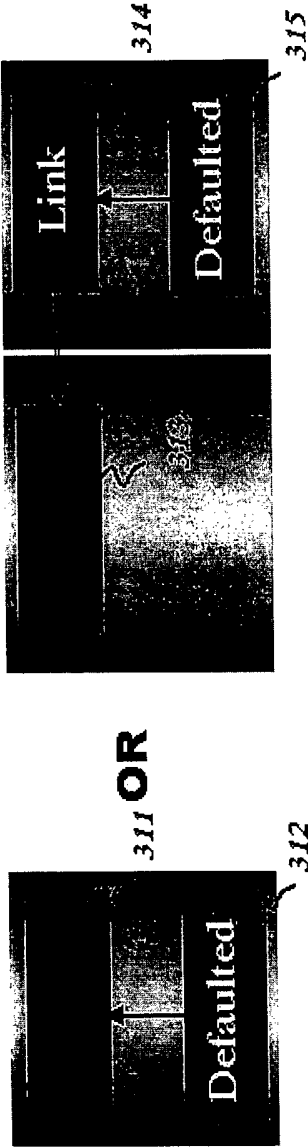


FIG 32

Question Source: Defaulted (cont.)

Example 1

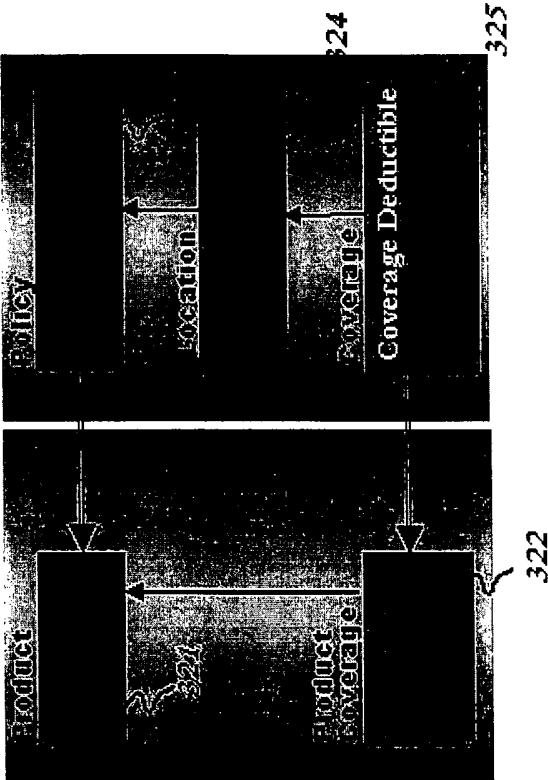


FIG 33

Question Source: Defaulted (cont.)

Example 2

