

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 October 2007 (11.10.2007)

PCT

(10) International Publication Number
WO 2007/112949 A1

(51) International Patent Classification:
G06Q 10/00 (2006.01) **G06F 9/44** (2006.01)

(21) International Application Number:
PCT/EP2007/002835

(22) International Filing Date: 29 March 2007 (29.03.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/397,026 31 March 2006 (31.03.2006) US

(71) Applicant (for all designated States except US): **SAP AG**
[DE/DE]; Dietmar-Hopp-Allee 16, 69190 Walldorf (DE).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **PRIGGE, Uta**
[DE/DE]; Langengarten 16, 69124 Heidelberg (DE). **FREUND, Jens** [DE/DE]; Schwetzingen Strasse 132,
69124 Heidelberg (DE). **KÄTKER, Stefan** [DE/DE];
Gassenweg 2c, 69221 Dossenheim (DE).

(74) Agent: **SCHIUMA, Daniele**; Müller-Boré & Partner,
Grafinger Strasse 2, 81671 München (DE).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

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.

(54) Title: COMPOSITE APPLICATION MODELING

(57) Abstract: Methods and apparatus, including computer program products, to realize a software model are described. A user-centric modeling component is defined, the user-centric modeling component modeling an interactive software system and including a plurality of process steps, the plurality of process steps defining a business scenario. For each process step in the plurality of process steps one or more service requirements are identified, where a service requirement represents functionality necessary to accomplish a step. For each service requirement a process component capable of providing the service requirement is identified, the process component modeling software implementing a business process and defining a service operation for providing the service requirement and for interacting with other process components.



WO 2007/112949 A1

COMPOSITE APPLICATION MODELING

BACKGROUND

The subject matter of this patent application relates to modeling software systems, and more particularly to modeling the composition and interaction of components in a software system including user interfaces.

Enterprises are typically modifying and adapting enterprise software systems with increasing frequency. But enterprise software systems are generally large and complex. Such systems can require many different components, distributed across many different hardware platforms, possibly in several different geographical locations. Typical software systems may not be able to reduce this complexity for end users. In order to change the design, configuration and implement new parts of an enterprise software system, one is required to understand details of the system at varying levels, depending on his or her role in designing, managing or implementing the system.

For example, some individuals may grasp how a given business scenario works from a business person's perspective, but do not necessarily understand how to model a software architecture formally to realize the business scenario. Whereas others may have knowledge of a software model but no know how to adapt the model to a business scenario. Given a business scenario that needs to be realized, composite application modeling helps users to understand and adapt existing software systems. This includes modeling components with and without a user interface. It may be that only the user interface needs to be adjusted, not the underlying software components.

SUMMARY

In general, in one aspect, embodiments of the invention feature defining a user-centric modeling component, the user-centric modeling component modeling an interactive software system and including a plurality of process steps, the plurality of process steps defining a business scenario. For each process step in the plurality of process steps identifying one or more service requirements, where a service requirement represents functionality necessary to accomplish a step. For each service requirement, identifying a process component capable of providing the service requirement, the process component modeling software implementing a

business process and defining a service operation for providing the service requirement and for interacting with other process components.

These and other embodiments can optionally include one or more of the following features. The plurality of process steps are defined by a composite pattern. The one or more service requirements are defined by a service pattern. The plurality of services are defined by a service pattern. Defining a new process component to provide a service requirement. An existing process component can be modified to provide a service requirement. A detailed process step model can be defined, the detailed process step model including one or more decision points. A detailed process step model can be defined, the detailed process step model including one or more user interface screens, one or more user roles, and one or more departments. A detailed process step model can be defined, the detailed process step model including one or tasks to be accomplished by a department or user role. One or more process component interaction models can be generated with a plurality of identified process components. A process component models software implementing a distinct business process. The user-centric modeling component is part of a composite integration scenario; and the composite integration scenario comprises a plurality of logically associated process components to realize a business scenario. Each process component belongs to exactly one deployment unit; and a deployment unit characterizes independently operable software deployable on a separate computer hardware platform. A detailed process step model is defined, the detailed process step model including one or more of: a decision point, an event, a workflow, an assigned department or a role.

In general, in another aspect, embodiments of the invention feature defining a user-centric modeling component including a plurality of process steps, the plurality of process steps defining a business scenario. For each process step in the plurality of process steps identifying one or more service requirements for each step. For each service requirement identifying a process component capable of providing the service requirement, the process component defining a service interface for providing the service requirement and for interacting with other process components. And defining a detailed process step model, the detailed process step model including one or more decision points.

In general, in another aspect, embodiments of the invention feature displaying, in a first view, a composite catalog illustrating composite integration scenarios, each of the

composite integration scenarios characterizing software implementing a respective and distinct business scenario. Displaying, in a second view, a composite integration scenario illustrating a user-centric component and one or more process components, where a process component characterizes software implementing a respective and distinct business process.

5 And displaying, in a third view, a process step model corresponding to the user-centric component, the process step model illustrating one or more process steps in a business scenario.

These and other embodiments can optionally include one or more of the following features. A user-initiated selection of a first graphical user interface associated with a single composite integration scenario in the first view is received, and wherein the second view is displayed in response to the selection of the first graphical user interface element. A user-initiated selection of a second graphical user interface associated with the user-centric component in the second view is received, and wherein the third view is displayed in response to the selection of the second graphical user interface element. Displaying, in a fourth view, a process component model illustrating one or more process interfaces and a business object. Receiving a user-initiated selection of a third graphical user interface associated with a process component in the second view, and wherein the fourth view is displayed in response to the selection of the third graphical user interface element.

In general, in another aspect, embodiments of the invention feature a user interface; and a model design component coupled to the user interface and configured to perform operations comprising displaying, in a first view, a composite integration scenario catalog illustrating one or more composite integration scenarios, each of the composite integration scenarios characterizing software implementing a respective and distinct business scenario. Displaying, in a second view, a composite integration scenario illustrating a user-centric component and one or more process components, where a process component characterizes software implementing a respective and distinct business process. And displaying, in a third view, a process step model corresponding to the user-centric component, the process step model illustrating one or more process steps in a business scenario.

Particular embodiments of the invention can be implemented to realize one or more of the following advantages. A model provides modeling entities to represent aspects of a software system. Multiple views of the model are provided in a user interface. The model

views offer varying levels of detail, allowing users to focus on the information that is important for their task. Model entities can be reused and correspond to reusable software that implements functionality corresponding to the model entity. The model supports dynamic mapping between incompatible message formats. A model can incorporate external components. The models can be used to generate metadata, which can be stored in a repository and used in various downstream processes and tools.

Models can be used at specification and design time, reused for testing, reused for consulting purposes and reused again for documentation (for the developing company, independent software vendors, and customers). Since different parts of a software system can be used within composites, there can be different organizational departments that realize a composite. The models improve collaboration between these departments by providing people with different backgrounds a common language for interaction.

Moreover, the specification discloses a logical abstraction of how various software modules can interact to realize a business scenario. In particular, effective use can be made of process components as units of software reuse, to provide a design that can be implemented reliably in a cost effective way. Deployment units, each of which is deployable on a separate computer hardware platform independent of every other deployment unit, are capable of enabling a scalable design. Furthermore, service interfaces of the process components can define a pair-wise interaction between pairs of process components that are in different deployment units in a scalable manner.

The modeling methodology allows users to quickly adapt software systems. The models can be used to improve communication between different user groups within a software company. Composite patterns can be used to support and simplify modeling of common composite integration scenarios. If business functionality is missing, a new process component with new business object(s) can be modeled.

Some processes are automated and do not need a user interface. Others need user interfaces, for example if a user has to enter data. Combining the end user experience with the application integration leads to the topic of composite applications. Composite applications have a user interface. A composite application can define business activities covering one or many backend components in a user-centric way and provides independence

from underlying application systems. A composite application can guide one or many users through a set of process steps.

Models for composite modeling are used at specification and design time, reused for testing, reused for consulting purposes and reused again for documentation (for the
5 developing company, ISVs and customers). Since different parts of a software system are used within composites, there can be different departments that realize a composite. The models improve collaboration between these departments. Composite application modeling gives people with different backgrounds a common language to exchange information.

The details of one or more embodiments of the invention are set forth in the
10 accompanying drawings and the description below. Other features, aspects, and advantages of the invention will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a process flow diagram illustrating an approach for composite application modeling.

15 FIG. 2 is an illustration of a composite application modeling system.

FIG. 3 is an illustration of a process step model for a composite application.

FIG. 4 is a detailed process step model which adds decision logic, user interface screens, tasks, user roles and exposes communication in a process step model.

FIG. 5 is an illustration of a process step model, including service requirements for
20 process steps that represent functionality needed to accomplish the process steps.

FIG. 6 is an illustration of a process step model, paired with process components that can furnish needed service operations corresponding to the service requirements.

FIG. 7 is an illustration of a refined consumer model based on the process step model from FIG. 6 and illustrating service operations and their interfaces.

25 FIG. 8 is an illustration of a composite integration scenario.

FIG. 9 is an illustration of a process component interaction model.

FIG. 10A is an illustration of a process component model.

FIG. 10B is a further illustration of a process component model.

FIG. 11 is an illustration of a composite catalog.

30 FIG. 12 is an illustration of an approval composite pattern.

FIG. 13 is an illustration of an analyze-to-action composite pattern.

FIG. 14 is an illustration of a business object user interface composite pattern.

Like reference numbers and designations in the various drawings indicate like elements.

5

DETAILED DESCRIPTION

FIG. 1 is a flow diagram illustrating a technique 100 for composite application modeling. A process step model for a composite application component is defined from a business's point-of-view (step 102). This is done by a user such as a Business Analyst, for example. A process step model models an interactive software system that is built from
10 existing and, optionally, newly created components and includes a user interface. A process step model includes one or more process steps, which are logical steps in a business scenario. For example, two illustrated process steps are "Accept Customer Order" and "Fulfill Customer Order".

A detailed process step model is defined (step 104). The detailed process step model
15 is based on the process step model and augments process steps with branching logic and events. In addition, the detailed process step model can provide information about user interface involvement (e.g., which user interface screens are used) and department and user roles that accomplish a given process step.

One or more service requirements by one or more process steps in the process step
20 model are identified (step 106). A service requirement represents functionality that is needed to accomplish a process step. A model that shows the process steps and service requirements is termed a *consumer model*. Decision logic and other information is omitted so that the user creating the model can concentrate on process steps and service requirements.

As will be discussed below, a business scenario encompasses process component
25 models and interactions between them, which are required for realization of the business scenario. A process component model characterizes software implementing a respective and distinct business process. Each service requirement is paired with a process component model that is capable of fulfilling the service requirement (step 108). This can include using an existing process component model (step 110) or modifying an existing process component

model (step 112). Alternatively, a new process component model can be created to provide the service requirement (step 114).

When a composite application does not require the ability to persist information in a new business object, the composite application is termed *ultra-light*. In contrast, when a composite application requires the ability to persist information in a business object, the composite application is termed *heavy*. The process components paired with the service requirements are modeled in a *provider model*. A software implementation can be created from the defined models by generating service definitions, implementing the services, and developing the user interface.

FIG. 2 is an illustration of a composite application modeling system. An interactive graphical user interface (GUI) 204 allows users to interact with the modeling system to create, inspect and modify models. The GUI 204 can present a model in different views offering differing levels of detail. This allows users to focus on information that is appropriate to their role or the task at hand. The GUI allows users to drilldown to GUI mockups and actual GUIs for a composite application. A model design component 206 coupled to the GUI 204 provides one or more tools for modifying and manipulating one or more models. A repository 202 is capable of storing one or more models and associated information. By way of illustration, the repository can incorporate one or more files, databases, services, combinations of these, or other suitable means for providing persistent storage of model information.

FIG. 3 is an illustration of a process step model 300 for a composite application. Process steps, for example, can be defined by business analysts that understand how a given business scenario works from a business person's perspective down to processes and process steps, but do not necessarily understand how to model a software architecture formally to realize the business scenario. For example, a customer might have purchased a camera at a vendor. If the customer has a problem with the camera, the customer sends a given template with customer data and the camera data to the vendor. The data is validated and in case of correctness the accepted refunds coupon can be sent to the customer. In this illustration, a first process step 302 in a business scenario receives refunds data from a customer. A second process step 304 in the business scenario is for validating the refunds data. A third process step 306 creates a refunds coupon. Finally, a fourth process step 310 sends refunds to a

customer. Process steps can be defined at varying levels of granularity and although this figure illustrates four process steps, fewer or more process steps are possible. Process steps are generally performed in order beginning with the uppermost process step (e.g., 302) and following sequence arrows (e.g., 308). If a detailed process step model has been developed (see below), the decision logic therein can be omitted so that all possible process steps are listed in linear order.

FIG. 4 shows a detailed process step model 400 which adds decision logic, user interface screens, tasks, user roles and exposes communication in a process step model. Service requirements have been identified for process steps. A service requirement represents functionality needed to accomplish a given process step. By way of illustration, service requirements can be identified by software architects or developers. A service requirement is modeled as an invocation of a process component service operation. Process component service operations are described below.

In this illustration, the process step 302 for receiving refunds data from a customer has a receive refunds data service requirement 402. The process step 304 for validating refunds data has four service requirements: read delivery 404, read material 406, read customer 408, and read invoice 410. The process step 306 for creating a refunds coupon has a create refunds coupon service requirement 412. The process step 308 for sending refunds to a customer includes a sends refunds coupon service requirement 414.

The receive refunds data from Receive Refunds Data from customer process step 302 receives an electronic mail message 420 from a Refunds Processing at Customer process component 422 containing a refund request for a purchased product via the receive refunds data service requirement 402. Process step 304 is performed after which a decision point 438 is reached. A decision point represents conditional flow control logic (e.g., if-then, if-then-else, and combinations of these). If the refund data is correct, a data correct event 440 is generated and received by process step 306. Otherwise, a data incorrect event 442 is generated.

Process step 308 is performed after process step 306, assuming the data correct event 440 was generated. As part of the send refunds coupon to service requirement 414 in process step 308, an electronic mail message 426 is provided to the product data processing process component 422 to relay a refunds form.

Roles can be associated with process steps and organizations. For example, a vender organizational unit 416 is associated with process step 302 and an organizational unit role 418 is associated with the product data processing process step 422. User interface building blocks can be added to the model 400 to represent points of user interaction. A user interface building block can be a screen or part of a screen, for example, and can be reused in other models. A check user interface user interface block 430 is associated with the validate refunds data process step 304 and a refunds user interface block 432 is associated with process step 306.

In addition to what is described above, a detailed process step model can include operation requirements, interfaces, services, packages (which are logical groupings of process steps), forks, joins, iterations (e.g., which are typically done when something has been rejected instead of approved), forms, workflows shown by role assignments, involved departments (even different companies that are involved in a process). Other model entities are also possible.

FIG. 5 is an illustration of a process step model 300, including service requirements for process steps (described in FIG. 4) that represent functionality needed to accomplish the process steps. Since the process step model models services that can be consumed, this model is termed a “consumer model”. The process step model can be derived from the model in FIG 4 by omitting information that is not needed for allocation of the needed service operations and process components.

FIG. 6 is an illustration of a process step model 300, including service requirements paired with process components that have service operations corresponding to the service requirements. An architect who knows the existing process components can create this model, for example. Service requirements 402 and 414 are paired with process component Refunds Processing at Customer 604. Service requirements 404 and 406 are paired with process component Outbound Delivery 602. Service requirement 408 is paired with process component 422. Service requirement 410 is paired with process component 606. Finally, service requirement 412 is paired with process component 608. External process components, such as process component 604, can be shown to place the modeled process components in their operational context relative to another system, e.g., a system belonging to another company, such as a customer or other third party.

FIG. 7 is an illustration of a refined consumer model 700 based on the process step model 300 from FIG. 6. A refined consumer model exposes interfaces and service operations in the process components that are paired with service requirements. A service operation represents functionality, such as the creation of a customer quote or releasing a customer quote. For example, service operation 702 is paired with service requirement 404, service operation 704 is paired with service requirement 406, service operation 706 is paired with service requirement 408, service operation 708 is paired with service requirement 410, and service operation 710 is paired with service operation 412.

In addition, a process component defines one or more respective service interfaces for interacting with other process components. A service interface (or “interface”) is a named grouping of one or more service operations. An interface specifies offered (inbound service interface) or used (outbound service interface) functionality. A user-centric component calls synchronous services, so this model utilizes with inbound service interfaces.

A service operation belongs to exactly one process component. A process component generally has multiple service operations. A service operation is the smallest, separately-callable functionality of a process component. An operation modeled by a set of data types used as input, output, and fault parameters serves as a signature. A service operation can use multiple message types for inbound, outbound, or error messages.

Process components can be assigned to multiple deployment units so that each process component is included in exactly one deployment unit. Each deployment unit characterizes independently operable software that can be separately deployed on different hardware platforms. Each process component is entirely included in exactly one deployment unit. Context independent interactions among a plurality of process components are defined so that all communication and interaction between a process component in one deployment unit and any process component in any other deployment unit takes place through the respective process service interfaces (i.e., through operation) of the process components.

Separate deployment units can be deployed on separate physical computing systems and include one or more process components. For example, a physical system can be a cluster of computers having direct access to a common database. The process components of one deployment unit interact with those of another deployment unit only using messages passed through one or more data communication networks or other suitable communication

channels. Thus, a deployment unit software entity deployed on a platform belonging to *Company A* can interact with a deployment unit software entity deployed on a separate platform belonging to *Company B*, allowing for business-to-business communication. Or deployment units in different divisions of the same company can interact with each other.

5 More than one instance of a given deployment unit software entity can execute at the same time.

FIG. 8 shows an illustration of a composite integration scenario model 800. When process components have been associated with process steps and service requirements, this model can be created by a business analyst, architect or developer. A composite integration
10 scenario model (or “scenario”) includes a user-centric component and one or more process components that have been previously associated and the interactions between them. In addition, in the models described above, an architect determined if the needed business logic exists or if new business logic has to be provided. The new business logic will be incorporated in one or more new process components residing in a new deployment unit 812
15 (e.g., because a composite application can be shipped independently). In a composite integration scenario model, the new process components can be illustrated in the same color as the user-centric component, showing the most important new entities of a composite application.

A user-centric process component is only represented once in an integration scenario
20 model, even though the actual flow in the software system might invoke the same component multiple times. A scenario describes at a high level the potential direct or indirect (i.e., through one or more other components) interaction between components in one or more deployment units that are relevant to realize the business scenario. For example, a user-centric component 804 represents a user interface and consumes services of the existing
25 process components 604, 422, 606, 802, 806, 808 and the new composite process component 608. The new process component 608 gets data from one or more services of process component 602. Components of customer systems can be incorporated here such as 604 (e.g., a process component outside company). Different interactions can be distinguished such as service interactions, direct interactions (with proprietary technology), email and other
30 interactions, for example. Internal details of components are not described, nor are details of

process component interactions (e.g., service interfaces, service operations and messages). This is done in other models which can be accessed via drilldown in user interface 204.

The scenario 800 includes three deployment units: 810, 812 and 814. The deployment unit 810 includes the outbound delivery processing process component 602. The process component 602 in deployment unit 810 provides one or more services for the refunds processing process component 608. The refunds processing process component 608 was newly created to provide services required by a process step model of the user-centric component 804. Newly created process components that provide services to a user-centric component are included in their own deployment unit. In addition to utilizing one or more services provided by the process component 608, the user-centric component 804 uses the services of process components 422, 606, 802, 806, 808, in deployment unit 814, and external process component 604.

FIG. 9 is an illustration of a process component interaction model (PCIM) 900. PCIMs can be reused in different scenarios. A PCIM is a view of a model that incorporates relevant model entities associated with potential interaction between two process components (e.g., 602, 608). Service interfaces, process agents and business objects that are not relevant to the potential interaction are excluded. The PCIM 900 shows interactions between a Refunds Processing process component 608 and an Outbound Delivery Processing process component 602.

Service operations are described for purposes of exposition in terms of process agents. A process agent (or “agent”) is an optional modeling entity representing software that implements a service operation. Service operations can be implemented through other conventional techniques. Service operations (and hence agents) can be synchronous or asynchronous, and inbound or outbound.

Synchronous outbound service operations send synchronous request messages and process response messages. Synchronous inbound service operations respond to messages from synchronous outbound service operations. Synchronous communication is when a message is sent with the expectation that a response will be received promptly. Asynchronous communication comes with the expectation that a response will be provided by a separate service operation invoked at a later point in time.

An asynchronous outbound service operation is specific to a sending business object. If the asynchronous outbound service operation is triggering a new communication to another process component, it is specific for the triggered process component. However, the same asynchronous outbound process service operation can be used for two service operations
5 which are part of the same message choreography.

Inbound service operations are called after a message has been received by a process component. Based on the process component's business object's status, inbound service operations may initiate communication across deployment units, may initiate business-to-business (B2B) communication, or both by sending messages using well-defined services.

10 The Refunds Processing process component 608 includes an Refunds Coupon business object 906 which can use a Query Delivery of Refunds Coupon outbound process agent 908 to invoke a Query Delivery Date and Receiver service operation 910 included in a Query Delivery Out service interface 914 which causes a Simple Delivery Query message 916 to be sent to the Outbound Delivery Processing process component 602. The Refunds
15 Coupon business object 906 can use a Notify of Refunds Creation outbound process agent 932 to invoke a Notify of Delivery Creation service operation 912 included in service interface 926 which causes a Refunds Notification message 926 to be sent to the Outbound Delivery Processing process component 602.

The Outbound Delivery Processing process component 602 receives the Simple
20 Delivery Query message 916 by way of a Find Delivery by Date and Receiver service operation 920 included in service interface 930 which provides the message 916 to the synchronous Query Outbound Delivery service agent 922. The agent 922 assembles the queried data from the business object outbound delivery 924 and provides the data to the message Simple Delivery Query Response 940. The service operation 910 receives the data
25 and provides the information to the agent 908.

The service agent 922 changes the Outbound Delivery Object 924 based on the message 926. The Outbound Delivery Processing process component 602 receives the Refunds Notification message 926 by way of a Change Delivery Based on Refunds Processing service operation 922 included in service interface 928 which provides the
30 message 926 to the Change Outbound Delivery service agent 934. The service agent 934 creates or modifies the Outbound Delivery Object 924 based on the message 926.

The message format of a message sent by an outbound service operation need not match the message format expected by an inbound service operation. If the message formats do not match, and the message is transformed, or mapped. Message mapping is indicated by interposition of an intermediary mapping model element between the source and the destination of the message in a PCM or a PCIM.

FIG. 10A is an illustration of a process component model (PCM) 1000. A PCM is a view of a model that incorporates the model entities associated with a particular process component. PCM 1000 is for refund processing and is the new process component of the composite. A PCM can also describe potential interactions between a process component and other process components in the same or in different deployment units. For example, the illustrated process component 1000 can interact with the Outbound Delivery Processing process component 602. Moreover, a PCM can describe interaction with external process components that are controlled by third parties or with User-centric component. This interaction is shown on the left of the model.

When business analysts and architects have assigned process components to process steps (Figure 6), the architects can search the PCM model in order to find out if the needed services already exist. If the needed services do not exist, they can be added. Here, a new process component has been created.

The PCM models service operations incorporated in a process component. For example, the inbound service operation 1004 called by the user-centric component, and outbound service operations 910 and 912. The arc 1010 connecting the outbound service operation 910 to the process component 602 represents that the outbound service operation 910 can invoke a service operation for the process component 602. (Similarly, arc 1012 connects outbound service operation 912 with process component 602.)

The PCM optionally models process agents (e.g., 1006, 908, 932) corresponding to the process component's service operations. For example, the inbound process agent 1006 models processing or responding to a message routed to inbound service operations 1004. The inbound process agent 1006, for example, will access and modify the business object 906 as part of the processing. The outbound process agents 908 and 932 can react to the state or a change in the state associated with the business object 906 by invoking service operations. The outbound process agent 908 can invoke the synchronous service operation

910 which will cause a message to be sent to process component 602. Likewise, the outbound process agent 932 can invoke the asynchronous service operation 912 which will also cause a message to be sent to process component 602.

If new services are consumed by the process component, these process components appear in the PCM. If new services are consumed by the user-centric component of the composite, the user-centric component does not appear. Service patterns are important in order to make sure developers use the same approach and cut (granularity) for services. Services are described below.

FIG. 10B is a further illustration of a PCM 1008. The PCM 1008 is for outbound delivery for the delivery of a camera to a customer. For the composite new services are needed, others have already existed before for other scenarios. The outbound delivery can interact, for example, with the new Refunds Processing process component 608 of the composite. The shaded region 1006 represents model components that already exist (i.e., that are not being modeled anew). The model 1008 includes service operations such as Change Delivery 922, Find Delivery by date and Receiver 920, Notify of Outbound Delivery 1026, and Request Invoicing 1022.

Inbound process agent 934 models processing or responding to a message routed to inbound service operation 922 and can, for example, access and modify business object 924 as part of the processing. Likewise, inbound process agent 922 models responding to a message routed to inbound service operation 920 and can access business object 924. Outbound process agent 1016 can react to the state or a change in the state associated with the business object 924 by invoking service operation 1026. Service operation 1026 will cause a message to be sent to external process component 1028. Outbound process agent 1018 can also react to the state or a change in the state associated with the business object 924 by invoking service operation 1022 which can cause a message to be sent to process component 1014.

FIG. 11 is an illustration of the composite catalog which presents available composite integration scenarios. From here users can navigate into the other models presented beforehand. Each view can present a different level of detail or emphasize a different aspect of the model. This allows for users to focus on the information that is important for carrying out their duties without being distracted by extraneous detail. In one variation, the GUI 204

allows users to “drill down” to increasing levels of model detail. For example, selection of a composite integration scenario icon 1102a in the composite integration scenario catalog 1102 can cause an associated composite integration scenario model to be presented. Likewise, selection of a graphical representation of a user-centric component in a composite integration scenario model can cause the associated process step model or detailed process step model to be presented. Selection of a graphical representation of a process component can cause an associated process component model to be presented.

In one implementation, the aforementioned graphical depictions can be presented singularly or in combination with each other in the GUI 204. Moreover, a given graphical depiction can present all of its underlying information or a portion thereof, while allowing other portions to be viewed through a navigation mechanism, e.g., user selection of a graphical element, issuance of a command, or other suitable means. Information can also be represented by colors in the display of model entities. For example, color can be used to distinguish types of business objects, types of process agents and types of interfaces.

Composite patterns and service patterns can be used to simplify the modeling of composite applications by providing reusable modeling components. A composite pattern characterizes common user interactions. TABLE 1 illustrates four such patterns.

COMPOSITE PATTERN	DESCRIPTION
Approval Composite	If a process is already established, a typical composite use case is to add approval steps. For example, information is retrieved and gathered for the approval of one or more designated users. After one user approves, the next user can approve. After approval completion the normal process foreseen within the software systems can proceed. <i>See</i> FIG. 12.
Collaboration Composite	Allows users to modify a business object to accomplish a task. For example, a quotation is created by a first user, a second user adds information to the quotation, and a third user finishes the quotation. The composite improves collaboration between the

	users via automating it using workflows.
Analyze-to-action Composite	Allows users to analyze aggregated information (reporting) and submit a request for users to perform a task based on the analysis. See FIG. 13.
Business Object User Interface Composite	Allows users to modify and display business objects in an easy way by furnishing a user interface layer. See FIG. 14.

TABLE 1

FIG. 12 is an illustration of an approval composite pattern 1200. Additional approval steps may have to be added to a composite. A composite pattern is a template that includes process steps and services that are typically needed to realize such a composite, for example:

5 An approval composite gathers data necessary to display for one or more people who have to approve a document such as a quotation. After approval the document is updated and subsequent processes can be started. For example the document might be sent to a customer.

The approval composite pattern 1200 includes a request process step 1202 for data gathering, a first guided procedure process step 1204, a second guided procedure process step 1206 for additional data processing necessary before the approval step (e.g., a form might be sent to a customer 1222 that sends additional data back 1226), an approve process step 1208, an update process step 1210 where the approved document gets a new status, and a follow up process step 1212 (e.g., sending the finalized document out to the customer).

The request process step 1202 has a service requirement 1214 for finding a business object and a service requirement 1216 for reading details of the business object. The first guided procedure setup process step 1204 has a service requirement 1218 for finding a business object by some additional criteria, and a second service requirement 1220 for sending a form to a customer. The service requirement 1220 uses a service operation defined by the process component 1222. The second guided procedure process step 1206 includes a service requirement 1226 for receiving a form from a customer.

The approve process step 1208 may include a service requirement or not (e.g., a user interface with a button 'approve' and a button 'rejected'). The update process step 1230 includes a service requirement 1230 for changing a business object (e.g., the status is changed to approved or rejected). Finally, the follow up process step 1212 has a service requirement 1232 for sending the completed form with the finalized document to a customer.

The service requirement 1232 utilizes a service operation defined by process component 1234. The service requirements 1214, 1216, 1218, and 1230 use operations defined by process components 1224 or 1228.

FIG. 13 is an illustration of an analyze-to-action composite pattern. for example, a sales agent might use an analytical application to analyze sales orders within different reports and drilldown possibilities. From here the agent might see that some action on a particular sales order is needed. This action might be changing the number of items within the sales order. Changing the sales order has to be done in an operational system, instead of an analytical system.

There are two process steps: analyze 1302 and execute 1304 for doing the changes. The analyze process step 1302 includes an analyze service requirement 1306 which utilizes a service operation from an analytics process component 1308. When the user has chosen an object on which actions in the operational system have to be done, an identifier of this particular object is save within a buffer. The execute process step 1304 includes a find business object by selection criteria service requirement 1310, which provides the object's identifier to the operational system and finds the object data therein. A read business object service requirement 1314 reads details of the business object which are needed for executing the necessary changes. When the user has done the needed changes on a user interface, a change business object service requirement 1316 is needed for changing the object in the operational system. The service requirements 1314 and 1316 utilize service operations defined by process component 1312.

FIG. 14 is an illustration of a business object user interface composite pattern. Often existing user interfaces are too complicated for occasional users. Many time business objects such as a new product or a customer or a customer invoice have to be created, changed or deleted by such occasional users. This composite pattern shows what process steps and services are typically necessary.

In a first screen the user enters some data. Then the system has to find out if the business object already exists or not. For this a request business object data process step 1402 includes a find help business object service requirement 1406 and a find business object process step 1408. Service requirement 1406 utilizes a service operation defined by a customer invoice processing process component and 1408 and service requirement 1410

utilizes a service provided by another process component 1412. Now the system has found out if a business object has to be created, changed or deleted. So the update or create business object process step 1404 includes service requirements 1414 for creation, 1416 and 1418 for changing and 1420 for deleting the business object. All service requirements 1414, 1416, 1418, 1420 utilize service operations defined by the process component 1412.

In addition to composite patterns which help business analysts to model their composite in a consumer model, there are service patterns that help architects and developers to model services the process components have to provide. A service pattern is used by composite applications in order to communicate with different components and can be created in a process component model or a process component interaction model. There are two types of service patterns: services consumed by users interfaces (typically synchronous, sometimes named A2X services) and services to communicate between process components (typically asynchronous, A2A or B2B services).

A2X service patterns can be used to represent communication between a user-centric component and one or more process components, and can be modeled within process component models. For example, a user-centric component can communicate with a process component that provides the user-centric component with operations on a business object, such as for creating, finding, modifying, deleting, accessing and triggering actions on a the business object. TABLE 2 illustrates three such patterns.

SERVICE PATTERN	DESCRIPTION
Collaboration Managing Services	Allows users to modify a business object to Create, change, delete business Objects. Naming convention examples: create <Business object> change<Business object> cancel <Business object> read <Business object>
Query Services	Find data in a system. Naming convention example: Find <Business Object> by <selection criteria> Example: Find Sales Order by status.
Action Services	Allows users to modify business objects via triggering actions

	such as approve, reject. This pattern can incorporate the triggering of an A2A service that changes other business objects in the underlying software components
--	--

TABLE 2

Asynchronous service patterns can be used to represent communication of new composite process component model(s) with existing process component model(s), and can be modeled with process component interaction models. TABLE 3 illustrates three such patterns. (These service patterns may also be synchronous.)

5

ASYNCHRONOUS SERVICE PATTERN	DESCRIPTION
Request/Confirmation	The basic pattern for transactional processing. A request is provided to a process component which immediately or later confirms the execution of an associated service operation to the requester.
Notification	A receiving process component accepts a business event notification without a reply being expected by the sending process component.
Query/Response	For data provisioning. Certain data is requested from and provided by a process component in a read-only manner.

TABLE 3

Embodiments of the invention and all of the functional operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the invention can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer-readable medium for execution by, or to control the operation of, data processing apparatus. The computer-readable medium can be a machine-readable storage device, a machine-readable

10

15

storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them.

The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data

from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio player, a Global Positioning System (GPS) receiver, to name just a few. Computer-readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

To provide for interaction with a user, embodiments of the invention can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

Embodiments of the invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the invention, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

5 While this specification contains many specifics, these should not be construed as limitations on the scope of the invention or of what may be claimed, but rather as descriptions of features specific to particular embodiments of the invention. Certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple
10 embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation
15 of a subcombination.

Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be
20 advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

25 Thus, particular embodiments of the invention have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results.

What is claimed is:

1. A method of defining an application, comprising:

defining a user-centric modeling component, the user-centric modeling component modeling an interactive software system and including a plurality of process steps, the plurality of process steps defining a business scenario;

identifying for each process step in the plurality of process steps one or more service requirements, where a service requirement represents functionality necessary to accomplish a step; and

identifying for each service requirement a process component capable of providing the service requirement, the process component modeling software implementing a business process and defining a service operation for providing the service requirement and for interacting with other process components.

2. The method of claim 1, where:

the plurality of process steps are defined by a composite pattern.

3. The method of claim 1 or 2, where:

the one or more service requirements are defined by a service pattern.

4. The method of anyone of claims 1 to 3, where:

the plurality of services are defined by a service pattern

5. The method of anyone of claims 1 to 4, further comprising:

defining a new process component to provide a service requirement.

6. The method of anyone of claims 1 to 5, further comprising:

modifying an existing process component to provide a service requirement.

7. The method of anyone of claims 1 to 6, further comprising:

defining a detailed process step model, the detailed process step model including one or more decision points.

8. The method of anyone of claims 1 to 7, further comprising:
defining a detailed process step model, the detailed process step model including one or more user interface screens, one or more user roles, and one or more departments.
9. The method of anyone of claims 1 to 8, further comprising:
5 defining a detailed process step model, the detailed process step model including one or tasks to be accomplished by a department or user role.
10. The method of anyone of claims 1 to 9, further comprising:
generating one or more process component interaction models with a plurality of identified process components.
- 10 11. The method of anyone of claims 1 to 10, where:
a process component models software implementing a distinct business process.
12. The method of anyone of claims 1 to 11, where:
the user-centric modeling component is part of a composite integration scenario; and
the composite integration scenario comprises a plurality of logically associated
15 process components to realize a business scenario.
13. The method of anyone of claims 1 to 12, where:
each process component belongs to exactly one deployment unit; and a deployment unit characterizes independently operable software deployable on a separate computer hardware platform.
- 20 14. The method of anyone of claims 1 to 13, further comprising:
defining a detailed process step model, the detailed process step model including one or more of: a decision point, an event, a workflow, an assigned department or a role.

15. A method of defining an application, comprising:

defining a user-centric modeling component including a plurality of process steps, the plurality of process steps defining a business scenario;

identifying for each process step in the plurality of process steps one or more service requirements for each step;

identifying for each service requirement a process component capable of providing the service requirement, the process component defining a service interface for providing the service requirement and for interacting with other process components; and

defining a detailed process step model, the detailed process step model including one or more decision points.

16. A method comprising

displaying, in a first view, a composite catalog illustrating composite integration scenarios, each of the composite integration scenarios characterizing software implementing a respective and distinct business scenario;

displaying, in a second view, a composite integration scenario illustrating a user-centric component and one or more process components, where a process component characterizes software implementing a respective and distinct business process; and

displaying, in a third view, a process step model corresponding to the user-centric component, the process step model illustrating one or more process steps in a business scenario.

17. The method of claim 16, further comprising:

receiving a user-initiated selection of a first graphical user interface associated with a single composite integration scenario in the first view, and wherein the second view is displayed in response to the selection of the first graphical user interface element.

18. The method of claim 16 or 17, further comprising:

receiving a user-initiated selection of a second graphical user interface associated with the user-centric component in the second view, and wherein the third view is displayed in response to the selection of the second graphical user interface element.

19. The method of anyone of claims 16 to 18, further comprising:

displaying, in a fourth view, a process component model illustrating one or more process interfaces and a business object.

20. The method of anyone of claims 16 to 19, further comprising:

5 receiving a user-initiated selection of a third graphical user interface associated with a process component in the second view, and wherein the fourth view is displayed in response to the selection of the third graphical user interface element.

21. A system, comprising:

10 a user interface; and a model design component coupled to the user interface and configured to perform operations comprising:

displaying, in a first view, a composite integration scenario catalog illustrating one or more composite integration scenarios, each of the composite integration scenarios characterizing software implementing a respective and distinct business scenario;

15 displaying, in a second view, a composite integration scenario illustrating a user-centric component and one or more process components, where a process component characterizes software implementing a respective and distinct business process; and

displaying, in a third view, a process step model corresponding to the user-centric component, the process step model illustrating one or more process steps in a business scenario.

22. A computer program product, encoded on a computer-readable medium, operable to cause data processing apparatus to perform operations comprising:

displaying, in a first view, a composite integration scenario catalog illustrating one or more composite integration scenarios, each of the composite integration scenarios

5 characterizing software implementing a respective and distinct business scenario;

displaying, in a second view, a composite integration scenario illustrating a user-centric component and one or more process components, where a process component characterizes software implementing a respective and distinct business process; and

10 displaying, in a third view, a process step model corresponding to the user-centric component, the process step model illustrating one or more process steps in a business scenario.

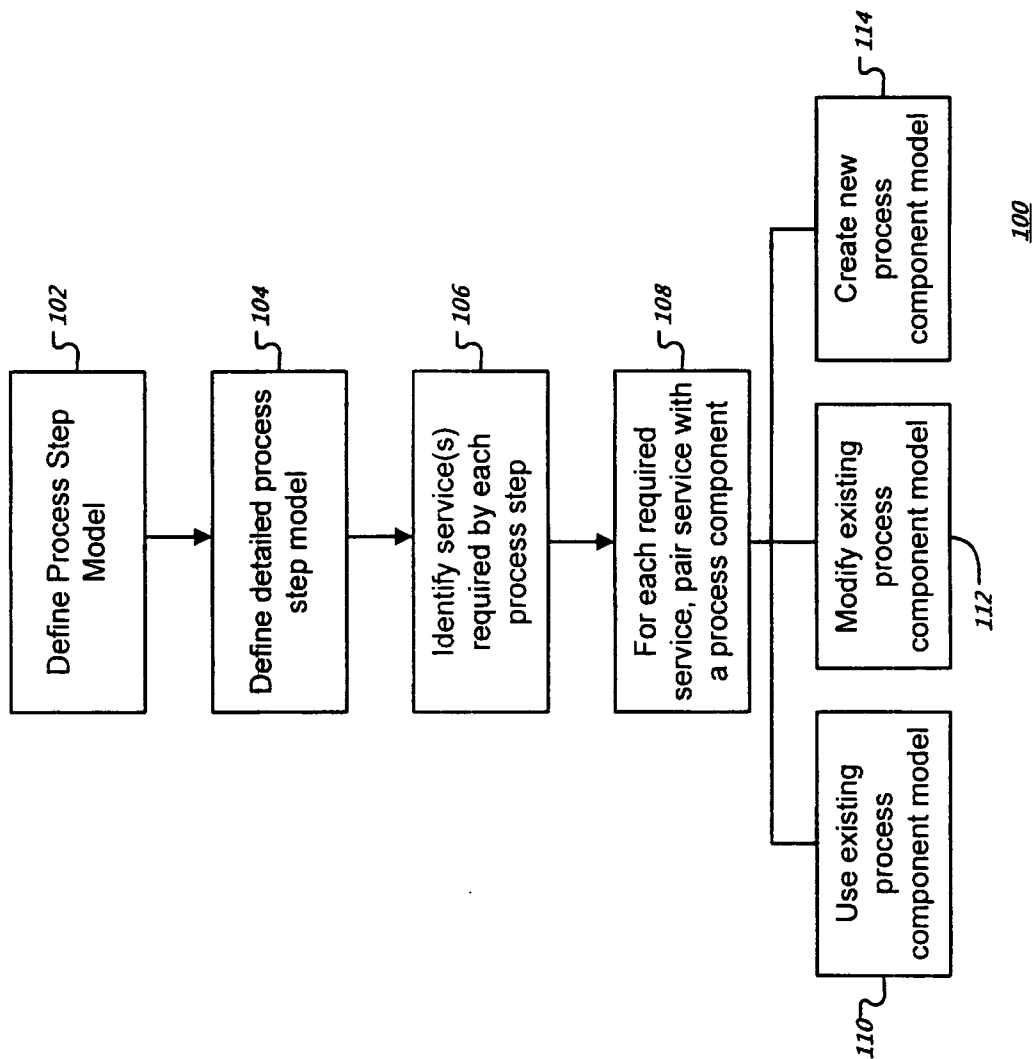
23. The computer program product of claim 22, operation to cause the data processing apparatus to perform further operations comprising:

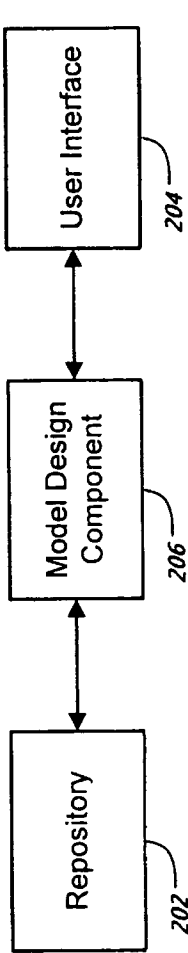
15 receiving a user-initiated selection of a first graphical user interface associated with a single composite integration scenario in the first view, and wherein the second view is displayed in response to the selection of the first graphical user interface element.

24. The computer program product of claim 22 or 23, operation to cause the data processing apparatus to perform further operations comprising:

20 receiving a user-initiated selection of a second graphical user interface associated with the user-centric component in the second view, and wherein the third view is displayed in response to the selection of the second graphical user interface element.

1/15

**FIG. 1**



200

FIG. 2

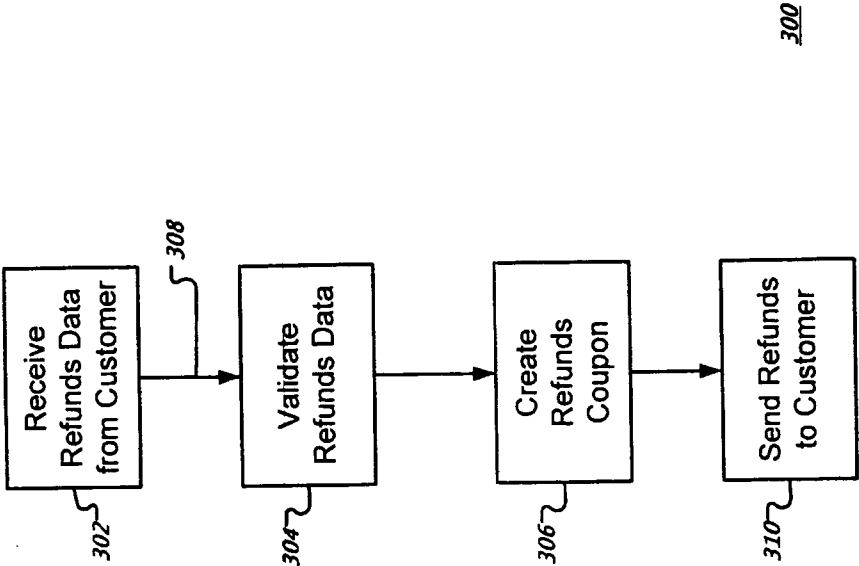


FIG. 3

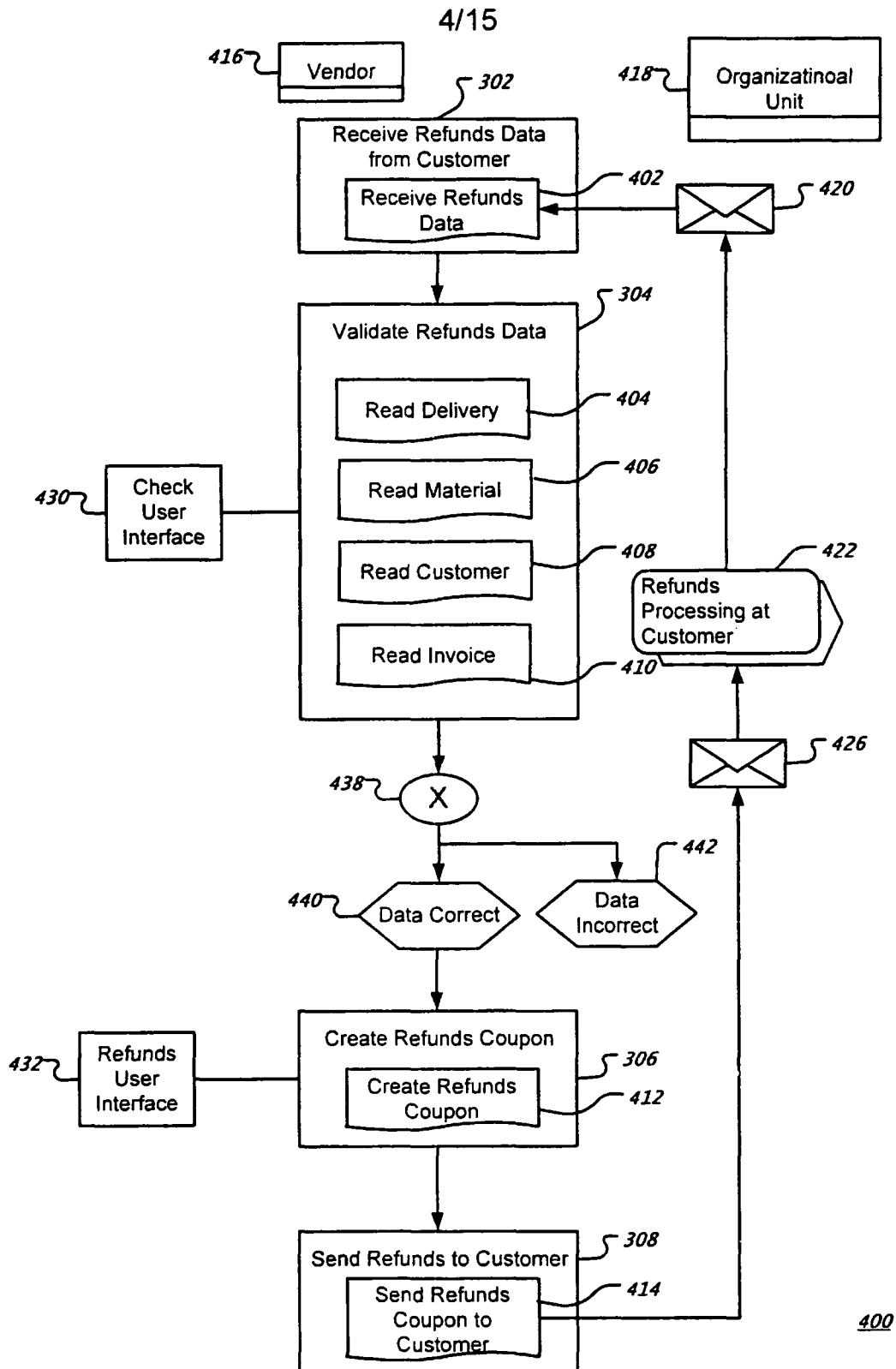
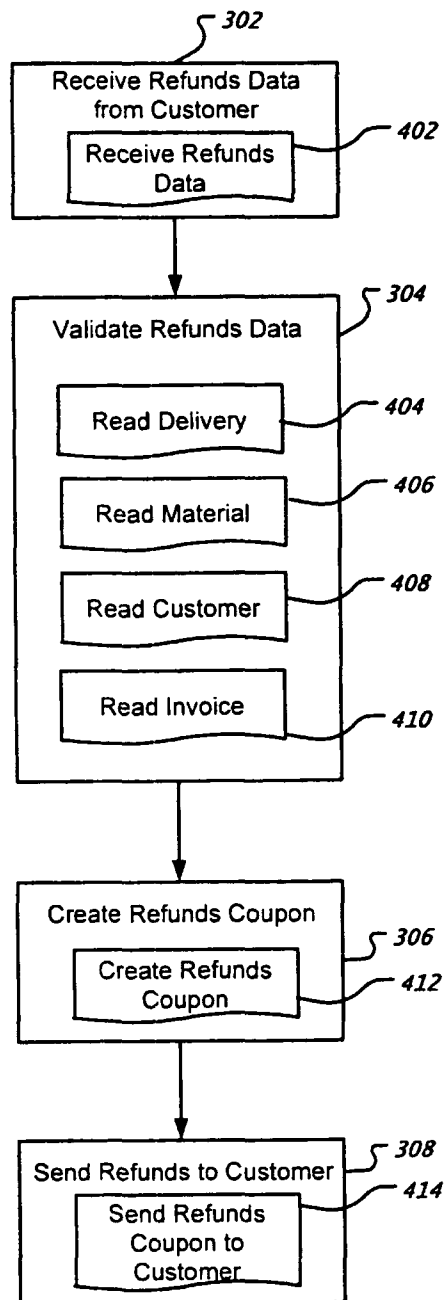
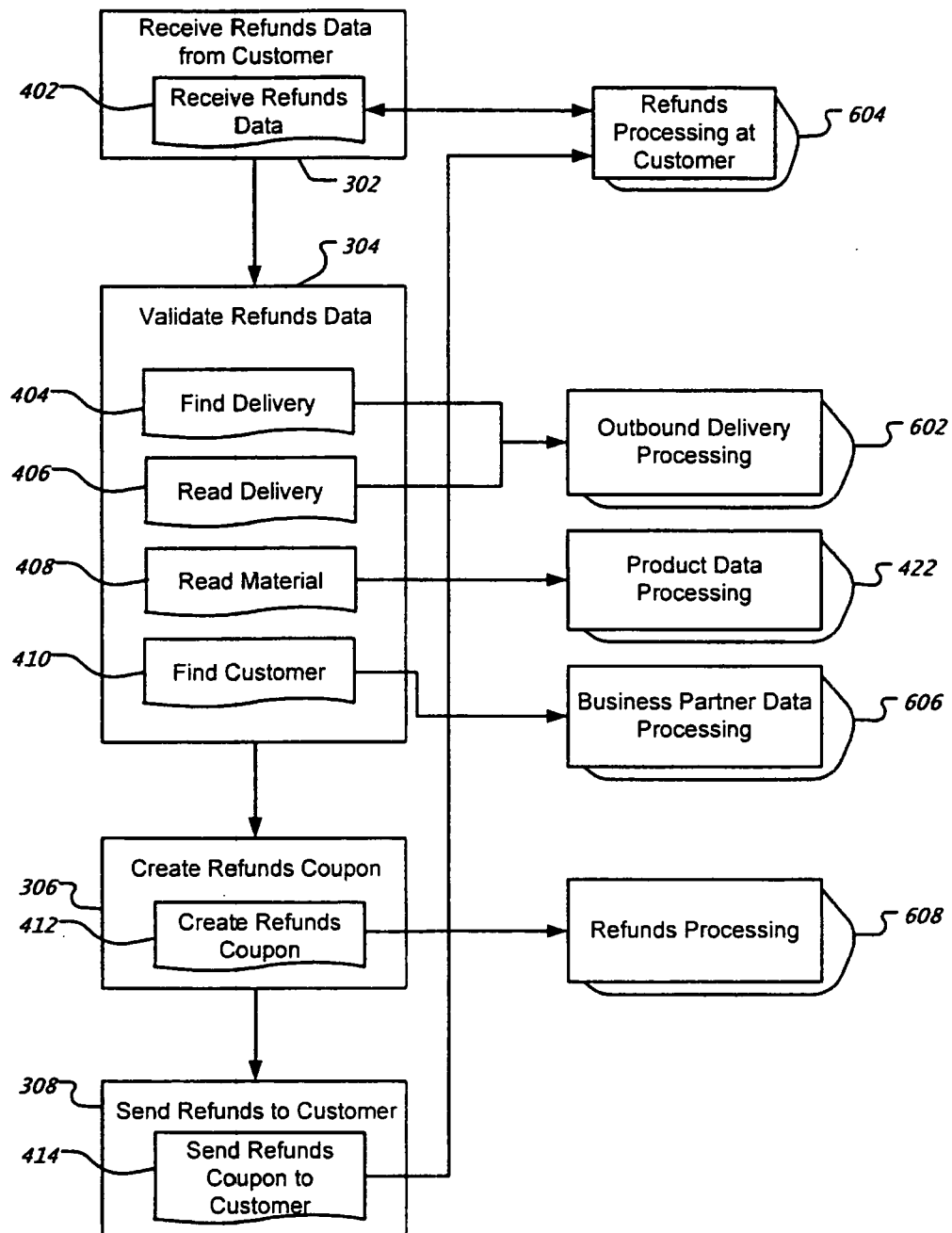


FIG. 4

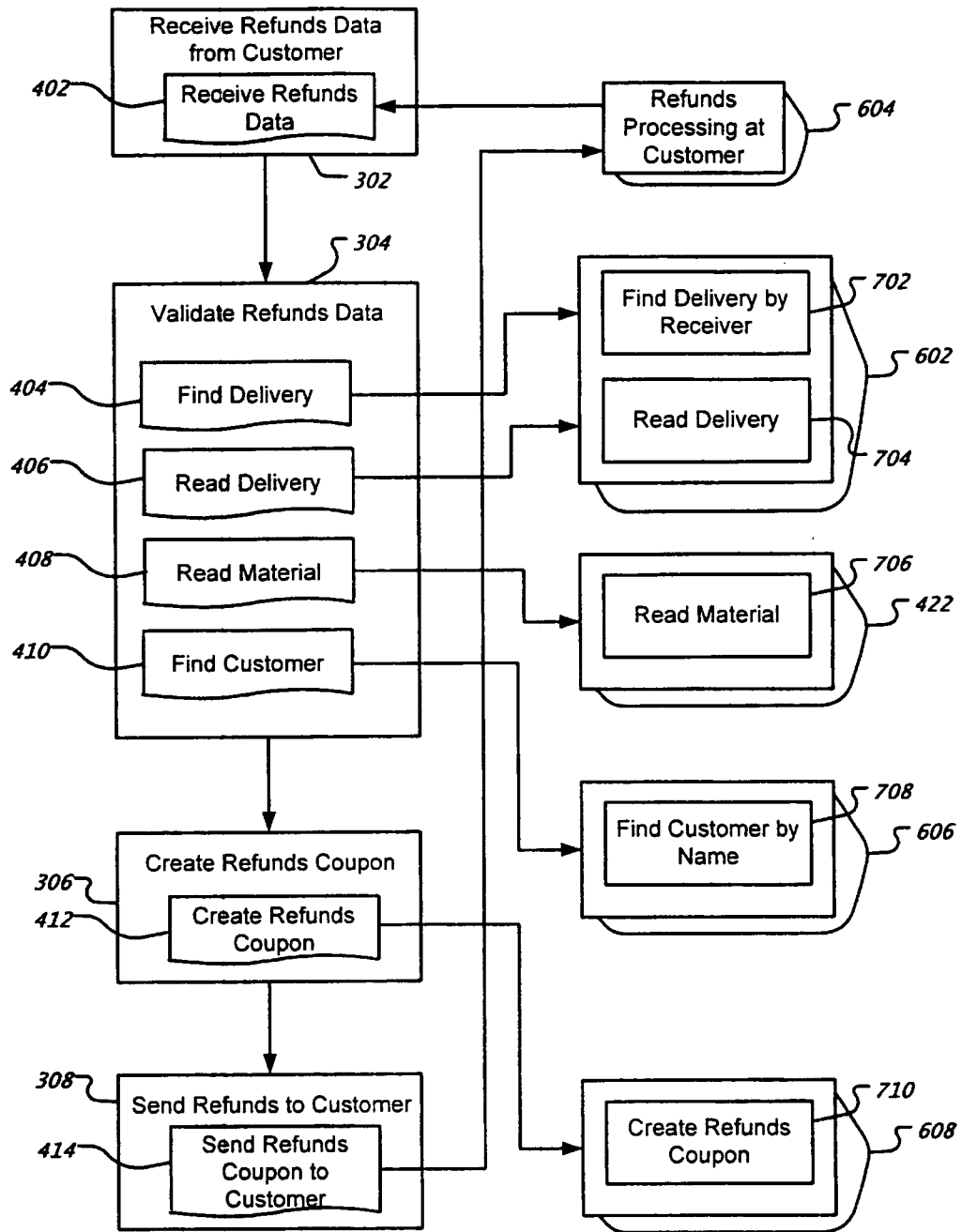
5/15

300**FIG. 5**

6/15

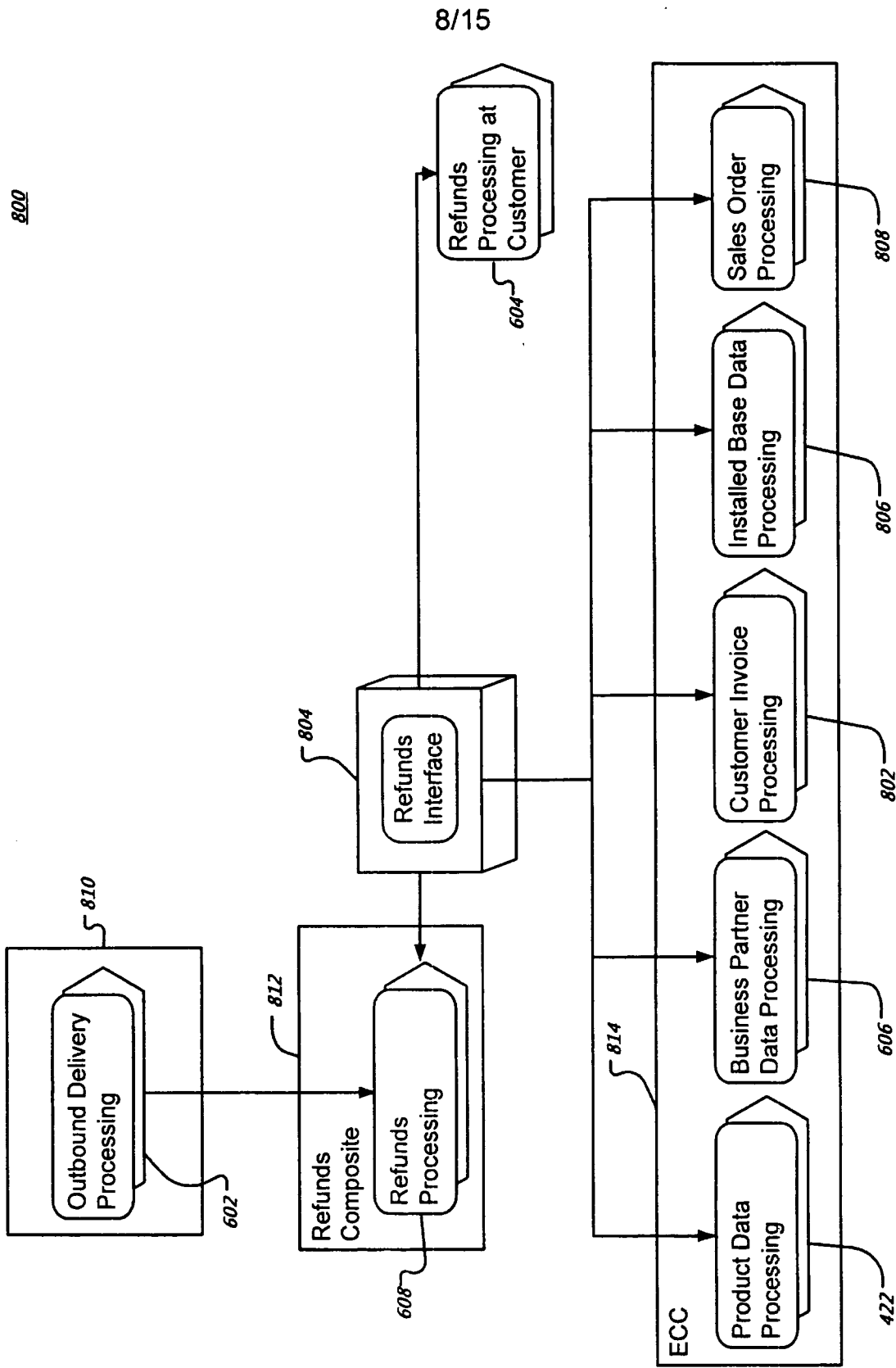
300**FIG. 6**

7/15

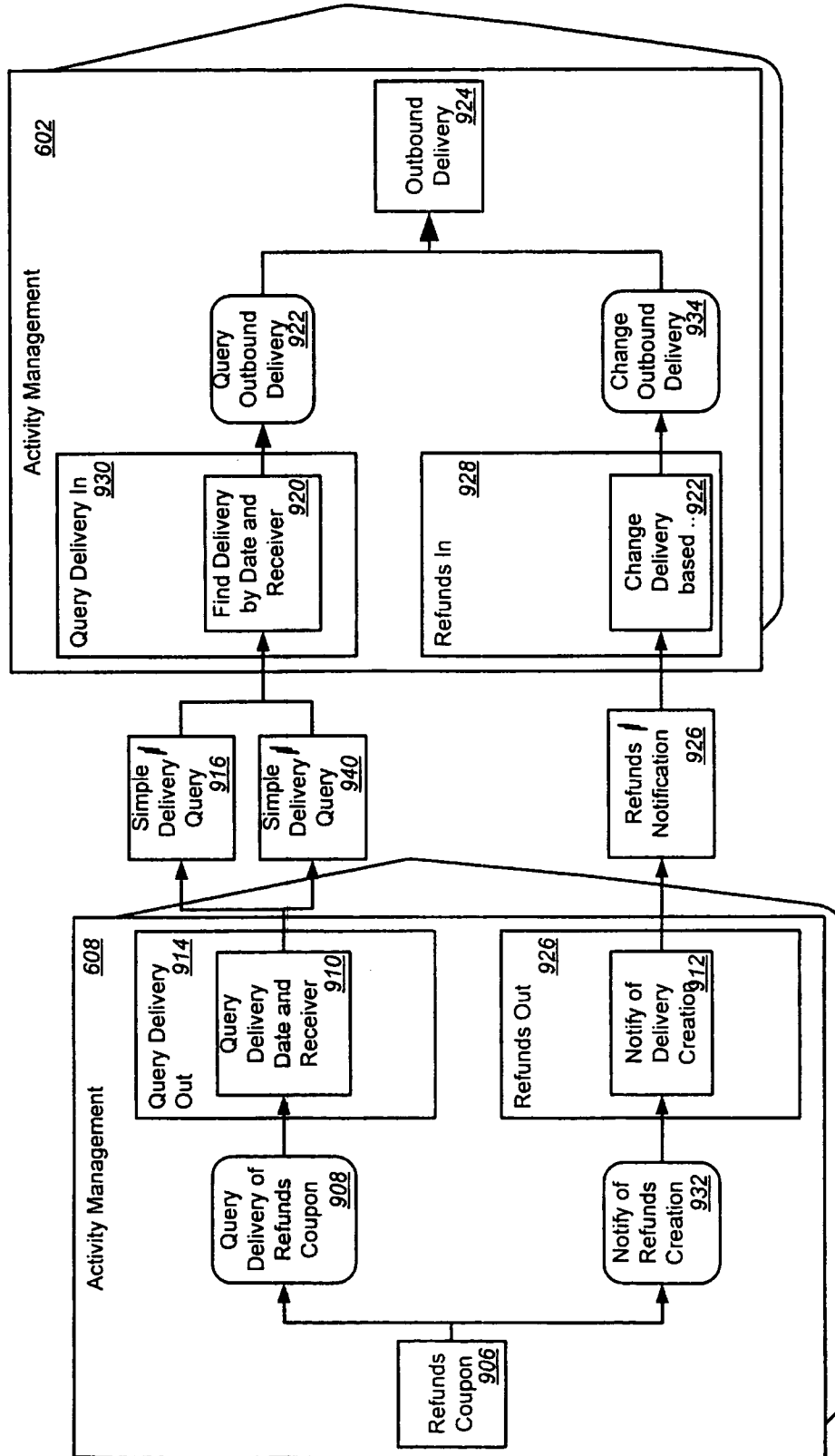


700

FIG. 7



9/15



900

FIG. 9

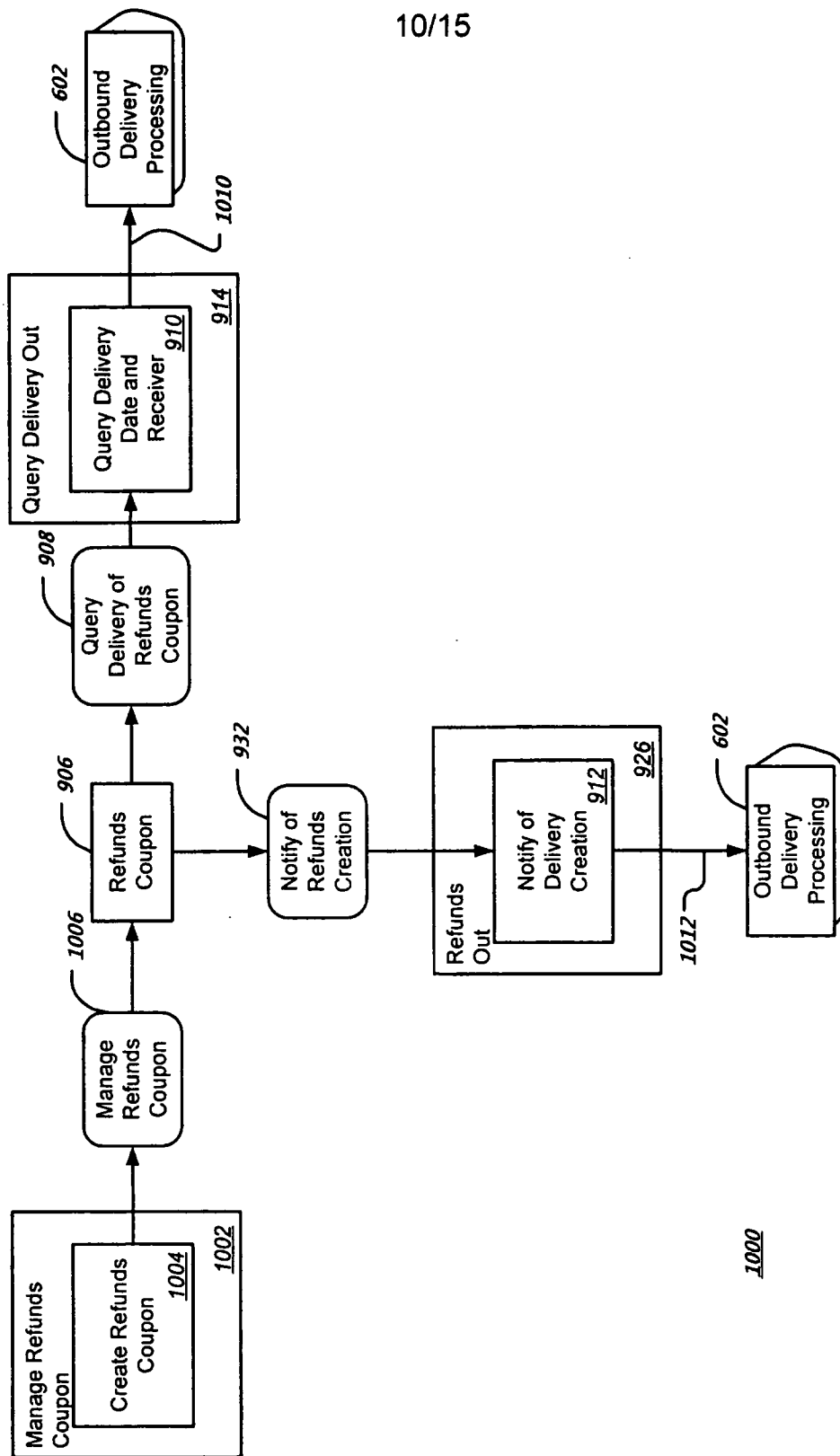


FIG. 10A

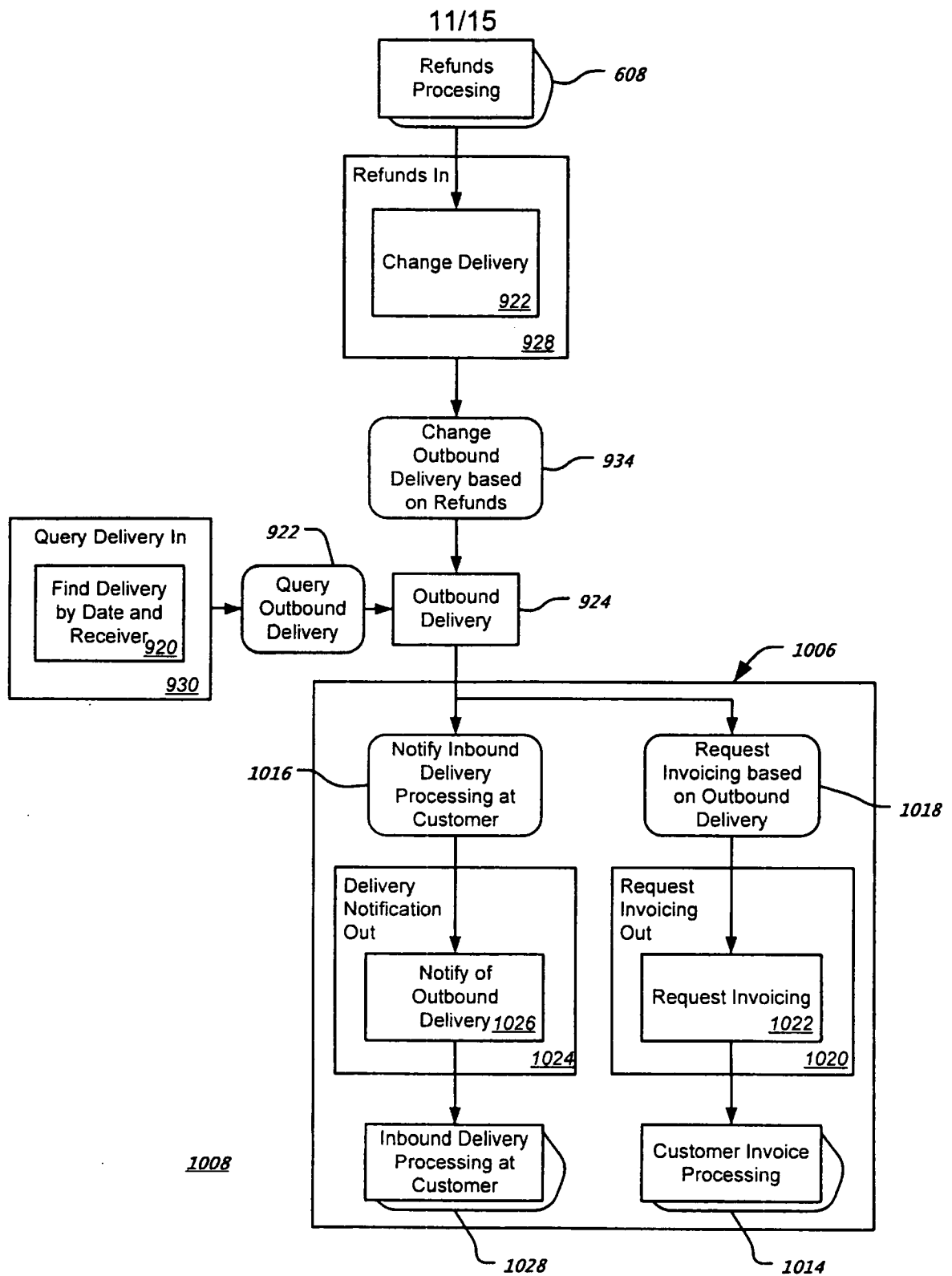


FIG. 10B

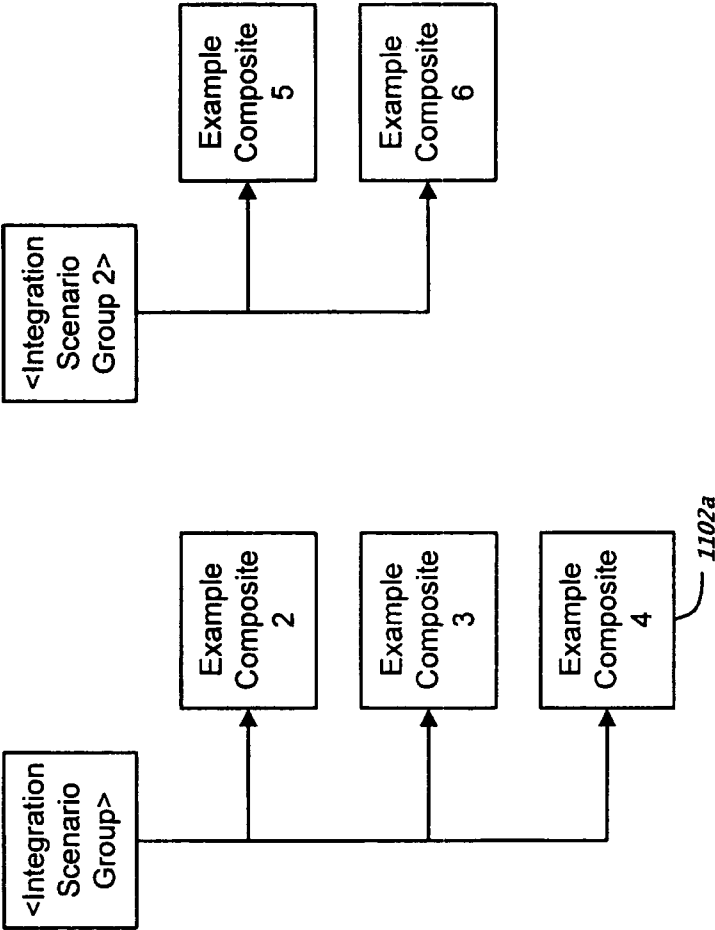


FIG. 11

1102

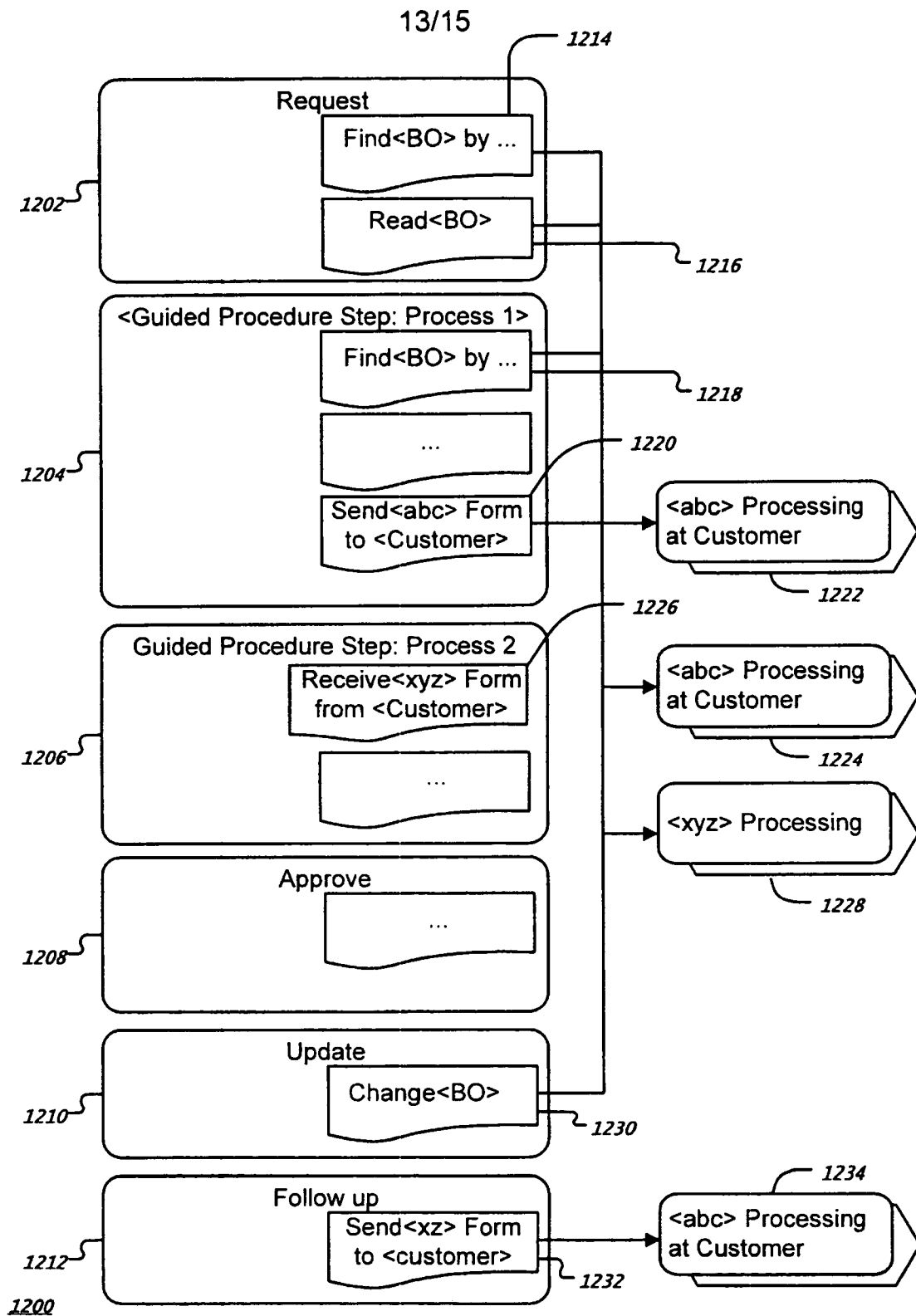
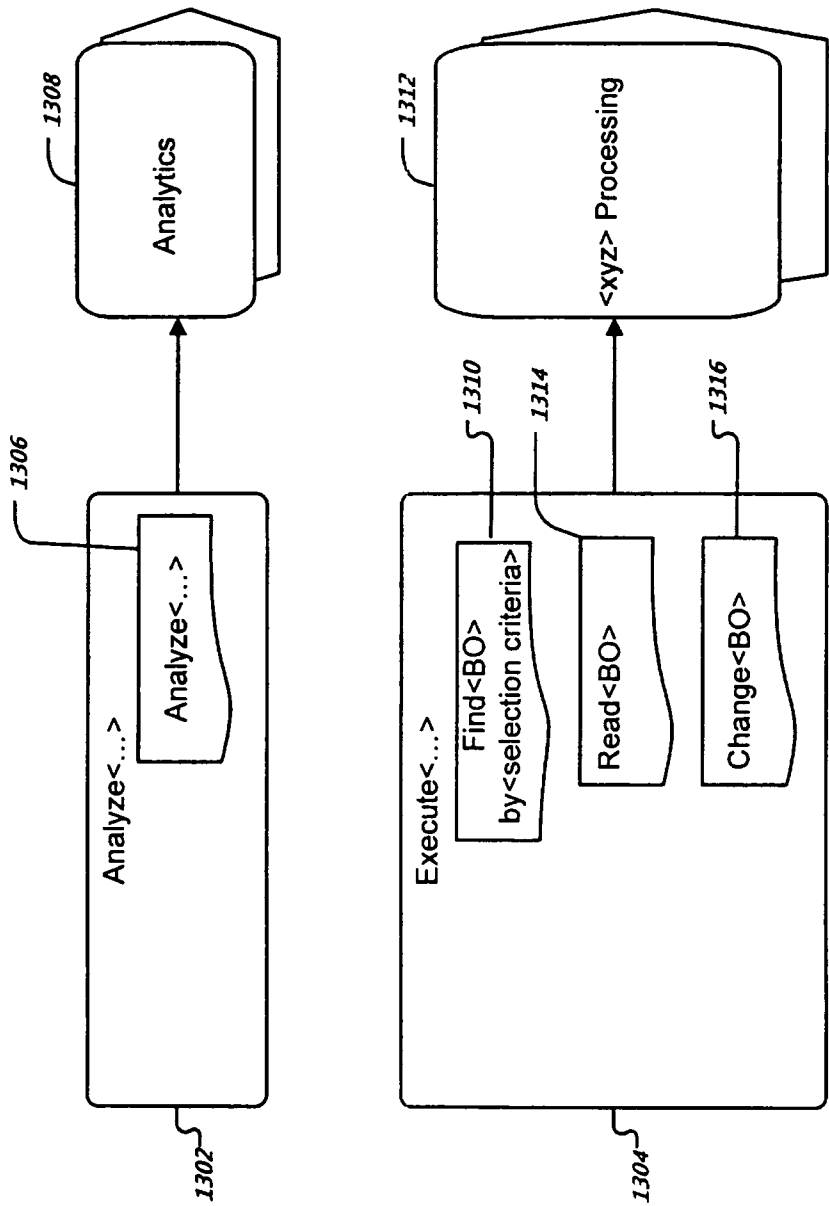


FIG. 12



1300

FIG. 13

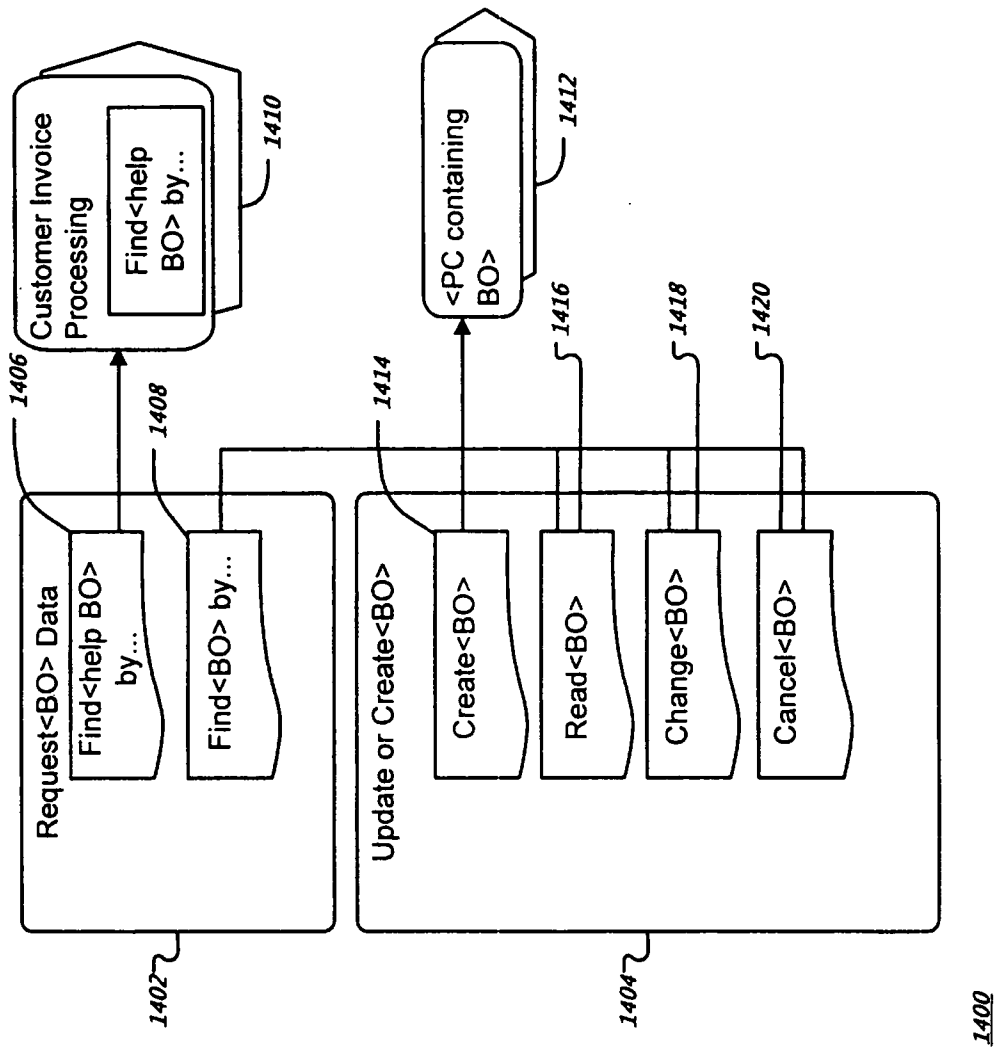


FIG. 14

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2007/002835

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06Q10/00 G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06Q G06F

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

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

EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2004/083984 A2 (SAP AG [DE]; AIGNER WERNER [DE]; ZIMMERMANN JUERGEN [DE]; BERINGER JOE) 30 September 2004 (2004-09-30) the whole document	21
X	WO 2005/114381 A (GT SOFTWARE INC [US]) 1 December 2005 (2005-12-01) the whole document	21
X	US 2005/144226 A1 (PUREWAL STEVEN [IN]) 30 June 2005 (2005-06-30) the whole document	21
	-/-	

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

Z document member of the same patent family

Date of the actual completion of the international search

25 July 2007

Date of mailing of the international search report

09/08/2007

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Corcoran, Patrick

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2007/002835

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	VERGIL TECHNOLOGY (P) LTD.: "Vergil Composite Application Builder Suite" PRODUCT DATASHEET, [Online] 2003, pages 1-5, XP007902689 ISSN: 3 Retrieved from the Internet: URL: http://www.webservicesmall.com/docs/VC_AB_datasheet.pdf [retrieved on 2007-07-23] the whole document	21
A	SAP AG: "DESIGNING CROSS SOLUTIONS" SAP XAPPS, [Online] September 2003 (2003-09), pages 1-2, XP007902686 US Retrieved from the Internet: URL: http://www.sap.com/belux/platform/networker/pdf/BWP_CAF.pdf [retrieved on 2007-07-23] the whole document	21
A	DAN WOODS: "Packaged Composite Applications: A Liberating Force for the User Interface" INTERNET CITATION, [Online] October 2003 (2003-10), XP002349437 Retrieved from the Internet: URL: http://www.sapdesignguild.org/editions/edition7/print_composite_applications.asp > [retrieved on 2005-10-17] the whole document	21
A	NORI A K ET AL: "Composite applications: process based application development" LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER VERLAG, BERLIN, DE, vol. 2444, August 2003 (2003-08), pages 48-53, XP002324956 ISSN: 0302-9743 the whole document	21
A	COWAN D D ET AL: "APPLICATION INTEGRATION: CONSTRUCTING COMPOSITE APPLICATIONS FROM INTERACTIVE COMPONENTS" SOFTWARE PRACTICE & EXPERIENCE, WILEY & SONS, BOGNOR REGIS, GB, vol. 23, no. 3, March 1993 (1993-03), pages 255-275, XP000655830 ISSN: 0038-0644 the whole document	21

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

Continuation of Box II.1

In particular, the subject matter of claims 1-20 and 22-24 is specified in such abstract terms that said claims, insofar as they can be understood, are considered to relate to schemes for performing mental acts in the sense of R.39(iii) PCT, presentations of information in the sense of R.39(v) PCT and computer programs in the sense of R.39(vi) PCT for implementing such mental acts and/or presentations of information.

To the extent that the subject-matter of said claims can be understood, it is considered that the International Searching Authority is not required to search the application in respect of claims 1-20 and 22-24 having regard to the provisions of R.39 PCT.

Continuation of Box II.2

Claims Nos.: 1-20,22-24

The wording of the claims does not allow a reliable determination of the technical features of the matter for which protection is sought contrary to the requirements deriving from Art. 6 and R. 6.3(a) PCT.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2007/002835

Box II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☒ Claims Nos.: 1-20, 22-24
because they relate to subject matter not required to be searched by this Authority, namely:
see FURTHER INFORMATION sheet PCT/ISA/210
2. ☒ Claims Nos.: 1-20, 22-24
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
see FURTHER INFORMATION sheet PCT/ISA/210
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest.
- ☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2007/002835

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2004083984 A2	30-09-2004	EP 1606710 A1 EP 1625468 A2 WO 2004084067 A1 US 2004187140 A1	21-12-2005 15-02-2006 30-09-2004 23-09-2004
WO 2005114381 A	01-12-2005	NONE	
US 2005144226 A1	30-06-2005	EP 1695205 A2 WO 2005048066 A2	30-08-2006 26-05-2005