(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

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

PCT

(10) **International Publication Number**
WO 2007/112956 A2

(51) **International Patent Classification:**
G06Q 30/00 (2006.01)

(21) **International Application Number:** PCT/EP2007/002842

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

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
11/396,250 30 March 2006 (30.03.2006) US

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

(72) **Inventor; and**
(75) **Inventor/Applicant** *(for US only)*: **PETER, Markus, A.** [DE/DE]; Viktoriastrasse 25, 68789 St. Leon-Rot (DE).

(74) **Agent: ROCKE, Carsten**; 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 declaration under Article 17(2)(a); without abstract; title not checked by the International Searching Authority*

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:** PROVIDING PRODUCT CATALOG SOFTWARE APPLICATION AS ENTERPRISE SERVICES

(57) **Abstract:**

PROVIDING PRODUCT CATALOG SOFTWARE APPLICATION AS ENTERPRISE SERVICES

BACKGROUND

This specification relates to data processing systems implemented on computers, and more particular to data processing systems providing services in the nature of web services.

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. Thus, the architecture of a large software application, i.e., what its components are and how they fit together, is an important aspect of its design for a successful implementation.

Web services are one technology for making the functionality of software applications available to other software, including other applications. A web service is a standards-based way of encapsulating the functionality of an application that other applications can locate and access. A service-oriented architecture is a distributed software model within which functionality is defined as independent web services. Within a service-oriented architecture, web services can be used in defined sequences according to business logic to form applications that enable business processes.

SUMMARY

This specification describes a services architecture design that provides enterprise services having product catalog functionality at the level of an enterprise application. Enterprise services are web services that have an enterprise-level business value.

In its various aspects, the invention can be embodied in systems, methods, and computer program products. For example, a system in one embodiment implements a services architecture design that provides enterprise services having product catalog functionality at the level of an enterprise application. The design includes a set of service operations, process components, and optionally deployment units. Suitable business objects are also described.

The subject matter described in this specification can be implemented to realize one or more of the following advantages. Effective use is made of process components as units of software reuse, to provide a design that can be implemented reliably in a cost effective way. Effective use is made of deployment units, each of which is deployable on a separate computer hardware platform independent of every other deployment unit, to provide a scalable design. Service interfaces of the process components define a pair-wise interaction between pairs of process components that are in different deployment units in a scalable way.

Details of one or more implementations of the subject matter described in this specification are set forth in the accompanying drawings and in the description below. Further features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a high-level view of a software architectural design and implementation of a suite of enterprise software services having product catalog functionality.

FIGS. 2A and 2B are block diagrams collectively showing a product catalog authoring process component.

FIGS. 3A and 3B are block diagrams collectively showing a product catalog publishing process component.

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

DETAILED DESCRIPTION

FIG. 1 illustrates a high-level view of a software architectural design, and of application software implementations of the design, that provides a suite of enterprise service operations, which can be organized into interfaces, having product catalog application functionality.

The elements of the architecture include the business object, the process component, the service operation (or simply, the operation), the service interface, the message, and the deployment unit. The elements can also include process agents and reuse service components. These will be generally described below.

In one implementation, the software is implemented to be deployed on an application platform that includes a foundation layer that contains all fundamental entities that can used from multiple deployment units. These entities can be process components, business objects or reuse service components. A reuse service component is a piece of software that is reused in different transactions. A reuse service component is used by its defined interfaces, which can be, e.g., local APIs (Application Programming Interfaces) or service interfaces.

The architectural design is a specification of a computer software application, and elements of the architectural design can be implemented to realize a software application that implements enterprise application service interfaces. The elements of the architecture are at times described in this specification as being contained or included in other elements; for example, a process component is described as being contained in a deployment unit. It should be understood, however, that such operational inclusion can be realized in a variety of ways and is not limited to a physical inclusion of the entirety of one element in another.

The architectural elements include the business object. A business object is a representation of a type of a uniquely identifiable business entity (an object instance) described by a structural model. Processes operate on business objects.

A business object represents a specific view on some well-defined business content. A business object represents content, and instances of business objects include content, which a typical business user would expect and understand with little explanation. Whether an object as a type or an instance of an object is intended by the term is generally clear from the context, so the distinction will be made explicitly only when necessary. Properly implemented, business objects are implemented free of redundancies.

Business objects are further categorized as business process objects and, master data objects, mass data run objects, dependent objects, and transformed objects. A master data object is an object that encapsulates master data (i.e., data that is valid for a period of time). A business process object, which is the kind of business object generally found in a process component, is an object that encapsulates transactional data (i.e., data that is valid for a point in time). A mass data run object is an application object that executes an algorithm for a particular mass data run. An instance of a mass data run object contains a particular set of selections and parameters. A mass data run object implements an algorithm that modifies, manages, and/or processes a large amount of data in multiple transactions, possibly but not necessarily with parallel processing. A dependent object is a business object used as a reuse part in another business object. A dependent object represents a concept that cannot stand by itself from a business point of view. Instances of dependent objects only occur in the context of a non-dependent business object. A transformed object is a transformation of multiple business objects for a well-defined purpose. It transforms the structure of multiple business objects into a common structure. A transformed object does not have its own persistency. The term business object will be used generically to refer to a business process object and a master data object, unless the context requires otherwise. Properly implemented, business objects are implemented free of redundancies.

The architectural elements also include the process component. A process component is a software package that realizes a business process and generally exposes its functionality as services. The functionality contains business transactions. A process component contains one or more semantically related business objects. Any business object belongs to no more than one process component.

Process components are modular and context-independent. That they are context-independent means that a process component is not specific to any specific application and is reusable. The process component is the smallest (most granular) element of reuse in the architecture.

The architectural elements also include the operation. An operation belongs to exactly one process component. A process component generally has multiple operations. Operations can be synchronous or asynchronous, corresponding to synchronous or asynchronous process agents, which will be described below. An operation is the smallest, separately-callable function, described by a set of data types used as input, output, and fault parameters, or some combination of them, serving as a signature. For convenience in supporting use of the operations supported by a system implementing elements of the design, such a system can optionally include a repository of service descriptions that includes a standards-based description of each of the supported service operations.

The architectural elements also include the service interface, which may be referred to simply as an interface. An interface is a named group of operations. Each operation belongs to exactly one interface. An interface belongs to exactly one process component. A process component might contain multiple interfaces. In one implementation, an interface contains only inbound or outbound operations, but not a mixture of both. One interface can contain both synchronous and asynchronous operations. All operations of the same type (either inbound or outbound) which belong to the same message choreography will preferably belong to the same interface. Thus, generally, all outbound operations to the same other process component are in one interface.

The architectural elements also include the message. Operations transmit and receive messages. Any convenient messaging infrastructure can be used. A message is information conveyed from one process component instance to another, with the expectation that activity will ensue. An operation can use multiple message types for inbound, outbound, or error messages. When two process components are in different deployment units, invocation of an operation of one process component by the other process component is accomplished by an operation on the other process component sending a message to the first process component.

The architectural elements also include the process agent. Process agents do business processing that involves the sending or receiving of messages. Each operation will generally have at least one associated process agent. A process agent can be associated with one or more operations. Process agents can be either inbound or outbound, and either synchronous or asynchronous.

Asynchronous outbound process agents are called after a business object changes, e.g., after a create, update, or delete of a business object instance.

Synchronous outbound process agents are generally triggered directly by a business object.

An output process agent will generally perform some processing of the data of the business object instance whose change triggered the event. An outbound agent triggers subsequent business process steps by sending messages using well-defined outbound services to another process component, which generally will be in another deployment unit, or to an external system. An outbound process agent is linked to the one business object that triggers the agent, but it is sent not to another business object but rather to another process component. Thus, the outbound process agent can be implemented without knowledge of the exact business object design of the recipient process component.

Inbound process agents are called after a message has been received. Inbound process agents are used for the inbound part of a message-based communication. An inbound process agent starts the execution of the business process step requested in a message by creating or updating one or multiple business object instances. An inbound process agent is not the agent of a business object but of its process component. An inbound process agent can act on multiple business objects in a process component.

Synchronous agents are used when a process component requires a more or less immediate response from another process component, and is waiting for that response to continue its work.

Operations and process components are described in this specification in terms of process agents. However, in alternative implementations, process components and operations can be implemented without use of agents using other conventional techniques to perform the functions described in this specification.

The architectural elements also include the deployment unit. A deployment unit includes one or more process components and, optionally, one or more business objects, that are deployed together on a single computer system platform. Conversely, separate deployment units can be deployed on separate physical computing systems. For this reason, a deployment unit boundary defines the limits of an application-defined transaction, i.e., a set of actions that have the ACID properties of atomicity, consistency, isolation, and durability. To make use of database manager facilities, the architecture requires that all operations of such a transaction be performed on one physical database; as a consequence, the processes of such a transaction must be performed by the process components of one instance of one deployment unit.

The process components of one deployment unit interact with those of another deployment unit using messages passed through one or more data communication networks or other suitable communication channels. Thus, a deployment unit deployed on a platform belonging one business can interact with a deployment unit software entity deployed on a separate platform belonging to a different and unrelated business, allowing for business-to-business communication. More than one instance of a given deployment unit can execute at the same time, on the same computing system or on separate physical computing systems.

This arrangement allows the functionality offered by a deployment unit to be scaled to meet demand by creating as many instances as needed.

Since interaction between deployment units is through service operations, a deployment unit can be replaced by other another deployment unit as long as the new deployment unit supports the operations depended upon by other deployment units. Thus, while deployment units can depend on the external interfaces of process components in other deployment units, deployment units are not dependent on process component interactions (i.e., interactions between process components involving their respective business objects, operations, interfaces, and messages) within other deployment units. Similarly, process components that interact with other process components or external systems only through messages, e.g., as sent and received by operations, can also be replaced as long as the replacement supports the operations of the original.

Interactions between process components that occur only within a deployment unit are not constrained to using service operations. These can be implemented in any convenient fashion.

In contrast to a deployment unit, the foundation layer does not define a limit for application-defined transactions. Deployment units communicate directly with entities in the foundation layer, which communication is typically not message based. The foundation layer is active in every system instance on which the application is deployed. Business objects in the foundation layer will generally be master data objects. In addition, the foundation layer will include some business process objects that are used by multiple deployment units. Master data objects and business process objects that should be specific to a deployment unit are preferably assigned to their respective deployment unit.

FIG. 1 illustrates a high-level view of a software architectural design and implementation of a suite of enterprise software services having product catalog functionality.

As shown in FIG. 1, a Catalog Authoring deployment unit 102 includes a Product Catalog Authoring process component 104, a Product Catalog master data object 106, a Product Catalog Update Run business object 108, a Product Catalog File Upload Run business object 110, a Product Catalog Duplication Run business object 112, a Product

Catalog Change List business object 114, a Product Catalog Update Method master data object 116, and a Product Catalog Publishing Sending Run business object 120.

Additionally, as shown in FIG. 1, a Catalog Publishing deployment unit 122 includes: a Product Catalog Publishing process component 124, a Published Product Catalog master data object 126, and a Published Product Catalog Update Run business object 128.

FIGS. 2A and 2B are block diagrams collectively showing the Product Catalog Authoring process component 104 (FIG. 1). For convenience in describing this process component, a number of other process components are shown in the figures; these other process components are not part of the process component being described. These other process components are a Purchasing Contract Processing process component 202, a Product Catalog Authoring at Supplier process component 204, a Product Catalog Publishing process component 124, and a Product Catalog Authoring at Customer process component 208. These other process components are used to represent software external to the process component in describing its interactions with the external software; however, while the external software can be implemented as such process components, this is not required.

A Maintain Catalog operation 210 sends a catalog maintenance request using a Maintain Product Catalog asynchronous inbound process agent 212 to update the Product Catalog master data object 106. For example, the operation 210 can send a request to update the Product Catalog master data object 106 if input is received from the Purchasing Contract Processing process component 202 or the Product Catalog Authoring at Supplier process component 204. The request can be to create a new product catalog, or to change or update an existing product catalog. The Maintain Catalog operation 210 is included in a Product Catalog Transmission Receiving In interface 214.

A Change Transmission Status operation 216 sends a product catalog transmission status change notification using a Change Product Catalog based on Published Product Catalog asynchronous inbound process agent 218 to update the Product Catalog master data object 106. For example, the operation 216 can send a change notification to update the Product Catalog master data object 106 if input is received from the Product Catalog Publishing process component 124. The change notification can be to update the status of a catalog publication transmission package sent out earlier.

A Change Publication Status operation 220 sends a product catalog publication status change using the Change Product Catalog based on Published Product Catalog asynchronous inbound process agent 218 to update the Product Catalog master data object 106. For example, the operation 220 can send a change notification to update the Product Catalog master data object 106 if input is received from the Product Catalog Publishing process component 124. The change notification can be to set the result status of an ongoing catalog publication transmission.

A Change Catalog based on Publication Cancellation operation 222 sends a product catalog change notification using the Change Product Catalog based on Published Product Catalog asynchronous inbound process agent 218 to update the Product Catalog master data object 106. For example, the operation 222 can send a change notification to update the Product Catalog master data object 106 if input is received from the Product Catalog Publishing process component 124. The change notification can be to update a catalog based on the cancellation of publication.

A Change Catalog based on Item Lock Status operation 224 sends a product catalog change notification using the Change Product Catalog based on Published Product Catalog asynchronous inbound process agent 218 to update the Product Catalog master data object 106. For example, the operation 224 can send a change notification to update the Product Catalog master data object 106 if input is received from the Product Catalog Publishing process component 124. The change notification can be to change a product catalog based on the change of status of a locked item within the catalog.

A Change Catalog based on Content Change Publication Status operation 226 sends a product catalog change notification using the Change Product Catalog based on Published Product Catalog asynchronous inbound process agent 218 to update the Product Catalog master data object 106. For example, the operation 226 can send a change notification to update the Product Catalog master data object 106 if input is received from the Product Catalog Publishing process component 124. The change notification can be to change a catalog based on a change of status of a catalog publication.

The Change Transmission Status operation 216, the Change Publication Status operation 220, the Change Catalog based on Publication Cancellation operation 222, the

Change Catalog based on Item Lock Status operation 224, and the Change Catalog based on Content Change Publication Status operation 226 are included in a Publishing In interface 228.

The Product Catalog master data object 106 can receive updated information and send the update into other components to perform further operations. As shown in FIG. 2B, multiple message agents can receive information from the Product Catalog master data object 106.

A Request Publication from Product Catalog to Product Catalog Publishing asynchronous outbound process agent 230 can invoke a Request Catalog Publication operation 232. For example, the outbound process agent 230 can send a request to publish a product catalog. The Request Publication from Product Catalog to Product Catalog Publishing asynchronous outbound process agent 230 can also invoke a Request Catalog Publication Cancellation operation 234. For example, the outbound process agent 230 can send a request to cancel publication of a product catalog. The Request Publication from Product Catalog to Product Catalog Publishing asynchronous outbound process agent 230 can also invoke a Request Catalog Item Lock operation 236. For example, the outbound process agent 230 can send a request to lock an item a product catalog. The Request Publication from Product Catalog to Product Catalog Publishing asynchronous outbound process agent 230 can also invoke a Request Catalog Publication Content Change operation 238. For example, the outbound process agent 230 can send a request to change the information for an item in a product catalog. The Request Catalog Publication operation 232, the Request Catalog Publication Cancellation operation 234, the Request Catalog Item Lock operation 236, and the Request Catalog Publication Content Change operation 238 are included in a Publishing Out interface 240.

A Notify of Publication from Product Catalog to Customer asynchronous outbound process agent 242 invokes a Notify of Catalog Update operation 244, which notifies another party (e.g., a customer) about a catalog publication. This can be a new catalog or an update to a previously published catalog. The Notify of Catalog Update operation 244 is included in a Transmission Sending Out interface 246. As an example, the operation 244 can then send a notification to the Product Catalog Authoring at Customer process component 208.

FIGS. 3A and 3B are block diagrams collectively showing the Product Catalog Publishing process component 124 (FIG. 1). For convenience in describing this process component, another process component is shown in the figures; this other process component is not part of the process component being described. This other process component is a Product Catalog Authoring process component 104. This other process component is used to represent software external to the process component in describing its interactions with the external software; however, while the external software can be implemented as such process components, this is not required.

A Maintain Published Product Catalog operation 304 can send a product catalog maintenance request using a Maintain Published Product Catalog asynchronous inbound process agent 306 to update the Published Product Catalog master data object 126. For example, the operation 304 can send a product catalog maintenance notification to update the Published Product Catalog master data object 126 if input is received from the Product Catalog Authoring process component 104. The maintenance notification can be to update the contents of a published catalog.

A Cancel Catalog Publication operation 308 can send a product catalog publication cancellation request using the Maintain Published Product Catalog asynchronous inbound process agent 306 to update the Published Product Catalog master data object 126. For example, the operation 308 can send a product catalog publication cancellation notification to update the Published Product Catalog master data object 126 if input is received from the Product Catalog Authoring process component 104. The maintenance notification can be to cancel the publication of a catalog.

A Lock Published Catalog Items operation 310 can send a product catalog item lock request using the Maintain Published Product Catalog asynchronous inbound process agent 306 to update the Published Product Catalog master data object 126. For example, the operation 310 can send a product catalog item lock notification to update the Published Product Catalog master data object 126 if input is received from the Product Catalog Authoring process component 104. The maintenance notification can be to place a lock on items within a published catalog.

A Change Published Catalog Content operation 312 can send a product catalog content change request using the Maintain Published Product Catalog asynchronous inbound process agent 306 to update the Published Product Catalog master data object 126. For example, the operation 312 can send a product catalog content change notification to update the Published Product Catalog master data object 126 if input is received from the Product Catalog Authoring process component 104. The maintenance notification can be to change the content of a published catalog.

The Maintain Published Product Catalog operation 304, the Cancel Catalog Publication operation 308, the Lock Published Catalog Items operation 310, and the Change Published Catalog Content operation 312 are included in a Publishing In interface 314.

The Published Product Catalog master data object 126 can receive updated information and send the update into other components to perform further operations. As shown in FIG. 3B, multiple process agents can receive information from the Published Product Catalog master data object 126.

A Notify of Product Catalog Publication Status asynchronous outbound process agent 316 can invoke a Notify of Publication Transmission Package Check operation 318. For example, the outbound process agent 316 can send a notification of a publication transmission package check. The Notify of Product Catalog Publication Status asynchronous outbound process agent 316 can also invoke a Confirm Catalog Publication operation 320. For example, the outbound process agent 316 can send a confirmation that a product catalog has been published. The Notify of Product Catalog Publication Status asynchronous outbound process agent 316 can also invoke a Confirm Catalog Publication Cancellation operation 322. For example, the outbound process agent 316 can send a confirmation that a product catalog publication has been cancelled. The Notify of Product Catalog Publication Status asynchronous outbound process agent 316 can also invoke a Confirm Catalog Item Lock operation 324. For example, the outbound process agent 316 can send a request to lock an item in a product catalog. The Notify of Product Catalog Publication Status asynchronous outbound process agent 316 can also invoke a Confirm Catalog Publication Content Change operation 326. For example, the outbound process agent 316 can send a request to change the information for an item in a product catalog. The Notify of Publication Transmission

Package Check operation 318, the Confirm Catalog Publication operation 320, the Confirm Catalog Publication Cancellation operation 322, the Confirm Catalog Item Lock operation 324, and the Confirm Catalog Publication Content Change operation 326 are included in a Publishing Out interface 328.

The subject matter described in this specification 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 structural means disclosed in this specification and structural equivalents thereof, or in combinations of them. The subject matter described in this specification can be implemented as one or more computer program products, i.e., one or more computer programs tangibly embodied in an information carrier, *e.g.*, in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, *e.g.*, a programmable processor, a computer, or multiple computers. A computer program (also known as a program, software, software application, 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. A program can be stored in a portion of a file that holds other programs or data, 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 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 executing 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. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, 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, the subject matter described in this specification 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.

The subject matter described in this specification can be implemented in a computing system that includes a back-end component (*e.g.*, a data server), a middleware component (*e.g.*, an application server), or 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 subject matter described herein), or any combination of such back-end, middleware, and 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.

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 an exemplification of preferred embodiments of the invention. Certain features that are described in this specification in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment may also be provided in multiple 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 of a subcombination.

The subject matter has been described in terms of particular variations, but other variations can be implemented and 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. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous. Other variations are within the scope of the following claims.

What is claimed is:

CLAIMS

1. A system comprising:

a computer system comprising one or more hardware platforms for executing computer software; and

computer software deployed on the computer system, the computer software implementing a plurality of service operations, the service operations comprising:

a maintain catalog operation, the operation operable to send a notification to update a product catalog;

a change transmission status operation, the operation operable to send a notification to change the transmission status of a product catalog;

a change publication status operation, the operation operable to send a notification to change the publication status of a product catalog;

a change catalog based on publication cancellation operation, the operation operable to send a notification to update a product catalog based on a publication cancellation;

a change catalog based on item lock status operation, the operation operable to send a notification to update a product catalog based on a locked item;

a change catalog based on content change publication status operation, the operation operable to send a notification to update a product catalog based on a catalog change content publication status;

a request catalog publication operation, the operation requesting that a product catalog be published;

a request catalog publication cancellation operation, the operation requesting a product catalog publication be cancelled;

a request catalog item lock operation, the operation requesting an item in a product catalog to be locked;

a request catalog publication content change operation, the operation requesting a content change in a product catalog publication;

a notify of catalog update operation, the operation operable to send a notification about an update to a product catalog;

a maintain published product catalog operation, the operation operable to send a notification to update a published product catalog;

a cancel catalog publication operation, the operation operable to send a notification to cancel the publication of a product catalog;

a lock published catalog items operation, the operation operable to send a notification to lock items contained in a published product catalog;

a change published catalog content operation, the operation operable to send a notification to change the content in a published product catalog;

a notify of publication transmission package check operation, the operation operable to send a notification about a catalog publication transmission package check;

a confirm catalog publication operation, the operation operable to send a notification to confirm the publication of a product catalog;

a confirm catalog publication cancellation operation, the operation operable to send a notification to confirm the cancellation of a product catalog publication;

a confirm catalog item lock operation, the operation operable to send a notification to confirm the lock of an item in a product catalog; and

a confirm catalog publication content change operation, the operation operable to send a notification to confirm the content change of a product catalog publication.

2. The system of claim 1, wherein the service operations are grouped into service interfaces, the service interfaces comprising:

a product catalog transmission receiving in interface that includes the maintain catalog service operation;

a publishing in interface that includes the change transmission status, the change publication status, the change catalog based on publication cancellation, the change catalog based on item lock status, and the change catalog based on content change publication status service operations;

a publishing out interface that includes the request catalog publication, the request catalog publication cancellation, the request catalog item lock, and the request catalog publication content change service operations;

a transmission sending out interface that includes the notify of catalog update service operation;

a publishing in interface that includes the maintain published product catalog, the cancel catalog publication, the lock published catalog items, and the change published catalog content service operations; and

a publishing out interface that includes the notify of publication transmission package check, the confirm catalog publication, the confirm catalog publication cancellation, the confirm catalog item lock, and the confirm catalog publication content change service operations.

3. The system of claim 1 or 2, wherein:

the computer software implementing the maintain catalog, the change transmission status, the change publication status, the change catalog based on publication cancellation, the change catalog based on item lock status, the change catalog based on content change publication status, the request catalog publication, the request catalog publication cancellation, the request catalog item lock, the request catalog publication content change, and the notify of catalog update operation is deployed on a first hardware platform; and

the computer software implementing the maintain published product catalog, the cancel catalog publication, the lock published catalog items, the change published catalog content, the notify of publication transmission package check, the confirm catalog publication, the confirm catalog publication cancellation, the confirm catalog item lock, and the confirm catalog publication content change operation is deployed on a second hardware platform.

4. The system of claim 3, wherein each of the first and second hardware platforms are distinct and separate from each other.

5. The system of claim 1, wherein:

the computer software implementing the maintain catalog, the change transmission status, the change publication status, the change catalog based on publication cancellation, the change catalog based on item lock status, the change catalog based on content change publication status, the request catalog publication, the request catalog publication cancellation, the request catalog item lock, the request catalog publication content change, and the notify of catalog update operation is deployable on a first hardware platform; and

the computer software implementing the maintain published product catalog, the cancel catalog publication, the lock published catalog items, the change published catalog content, the notify of publication transmission package check, the confirm catalog publication, the confirm catalog publication cancellation, the confirm catalog item lock, and the confirm catalog publication content change operation is deployable on a second hardware platform;

and wherein:

the first and second hardware platforms are distinct and separate from each other.

6. The system of any one of claims 1 to 5, wherein the computer software deployed on the computer system comprises:

a plurality of process components, each of the process components being a package of software deployed and executing on the computer system and implementing a respective and distinct business process, the plurality of process components including:

a product catalog authoring process component used to create and edit product catalogs, to control the quality of their contents, and to control their readiness for publication; and

a product catalog publishing process component used to make published product catalogs available electronically, and to provide a means to interactively search for and select products for use by other business processes;

and wherein:

the product catalog authoring process component implements the maintain catalog, the change transmission status, the change publication status, the change catalog based on publication cancellation, the change catalog based on item lock status, the change catalog based on content change publication status, the request catalog publication, the request catalog publication cancellation, the request catalog item lock, the request catalog publication content change, and the notify of catalog update service operations; and

the product catalog publishing process component implements the maintain published product catalog, the cancel catalog publication, the lock published catalog items, the change published catalog content, the notify of publication transmission package check, the confirm catalog publication, the confirm catalog publication cancellation, the confirm catalog item lock, and the confirm catalog publication content change service operations.

7. The system of any one of claims 1 to 6, wherein the computer software deployed on the computer system comprises:

a plurality of deployment units, each of the deployment units being a package of software packaged together to be deployed on a single physical hardware platform, the plurality of deployment units including:

a catalog authoring deployment unit that defines, edits and releases product catalogs; and

a catalog publishing deployment unit that provides released product catalogs for use in business processes;

and wherein:

the catalog authoring deployment unit implements the maintain catalog, the change transmission status, the change publication status, the change catalog based on publication cancellation, the change catalog based on item lock status, the change catalog based on content change publication status, the request catalog publication, the request catalog publication cancellation, the request catalog item lock, the request catalog publication content change, and the notify of catalog update service operations; and

the catalog publishing deployment unit implements the product catalog publishing process component implements the maintain published product catalog, the cancel catalog publication, the lock published catalog items, the change published catalog content, the notify of publication transmission package check, the confirm catalog publication, the confirm catalog publication cancellation, the confirm catalog item lock, and the confirm catalog publication content change service operations.

8. The system of any one of claims 1 to 7, further comprising:

a repository of service descriptions, the repository including a standards-based description of each of the plurality of service operations.

9. A computer program product encoded on a tangible machine-readable information carrier for implementing a plurality of services, the product comprising computer software operable to implement service operations on a computer system, the service operations comprising:

a maintain catalog operation, the operation operable to send a notification to update a product catalog;

a change transmission status operation, the operation operable to send a notification to change the transmission status of a product catalog;

a change publication status operation, the operation operable to send a notification to change the publication status of a product catalog;

a change catalog based on publication cancellation operation, the operation operable to send a notification to update a product catalog based on a publication cancellation;

a change catalog based on item lock status operation, the operation operable to send a notification to update a product catalog based on a locked item;

a change catalog based on content change publication status operation, the operation operable to send a notification to update a product catalog based on a catalog change content publication status;

a request catalog publication operation, the operation requesting that a product catalog be published;

a request catalog publication cancellation operation, the operation requesting a product catalog publication be cancelled;

a request catalog item lock operation, the operation requesting an item in a product catalog to be locked;

a request catalog publication content change operation, the operation requesting a content change in a product catalog publication;

a notify of catalog update operation, the operation operable to send a notification about an update to a product catalog;

a maintain published product catalog operation, the operation operable to send a notification to update a published product catalog;

a cancel catalog publication operation, the operation operable to send a notification to cancel the publication of a product catalog;

a lock published catalog items operation, the operation operable to send a notification to lock items contained in a published product catalog;

a change published catalog content operation, the operation operable to send a notification to change the content in a published product catalog;

a notify of publication transmission package check operation, the operation operable to send a notification about a catalog publication transmission package check;

a confirm catalog publication operation, the operation operable to send a notification to confirm the publication of a product catalog;

a confirm catalog publication cancellation operation, the operation operable to send a notification to confirm the cancellation of a product catalog publication;

a confirm catalog item lock operation, the operation operable to send a notification to confirm the lock of an item in a product catalog; and

a confirm catalog publication content change operation, the operation operable to send a notification to confirm the content change of a product catalog publication.

10. The product of claim 9, wherein the computer software comprises:

a plurality of process components, each of the process components being a package of software deployed and executing on the computer system and implementing a respective and distinct business process, the plurality of process components including:

a product catalog authoring process component used to create and edit product catalogs, to control the quality of their contents, and to control their readiness for publication; and

a product catalog publishing process component used to make published product catalogs available electronically, and to provide a means to interactively search for and select products for use by other business processes;

and wherein:

the product catalog authoring process component implements the maintain catalog, the change transmission status, the change publication status, the change catalog based on publication cancellation, the change catalog based on item lock status, the change catalog based on content change publication status, the request catalog publication, the request catalog publication cancellation, the request catalog item lock, the request catalog publication content change, and the notify of catalog update service operations; and

the product catalog publishing process component implements the maintain published product catalog, the cancel catalog publication, the lock published catalog items, the change published catalog content, the notify of publication transmission package check, the confirm catalog publication, the confirm catalog publication cancellation, the confirm catalog item lock, and the confirm catalog publication content change service operations.

11. The product of claim 9 or 10, wherein the computer software comprises:

a catalog authoring deployment unit that includes a product catalog, a product catalog update run, a product catalog file upload run, a product catalog duplication run, a product catalog change list, a product catalog update method, and a product catalog publishing sending run business object, wherein the product catalog business object is a master data object representing a structured directory of catalog items where each catalog item represents a product and provides information about it, the product catalog update run business object represents a run to update product catalogs based on product catalog update methods, the product catalog file upload run business object represents a run to update a product catalog based on an uploaded file, the product catalog duplication run business object represents a run to copy a product catalog in total or in parts, the product catalog change list business object represents a list of changes to a product catalog, the product catalog update method business object represents a master data object representing a set of rules controlling how products are cataloged in a product catalog and how to update the product catalog when cataloged products are changed, and the product catalog publishing sending run business object represents a run to trigger sending a product catalog for publication; and

a catalog publishing deployment unit that includes a published product catalog and a published product catalog update run business object, wherein the published product catalog business object is a master data object representing a version of a product catalog that has been released for access by or exchange with the target group of people for whom the product catalog has been tailored; and the published product catalog update run business object represents a run to update a product catalog based on an uploaded file.

12. The product of claim 10, wherein:

each of the plurality of process components is assigned to no more than one deployment unit among multiple deployment units, and each deployment unit is deployable on a separate computer hardware platform independent of every other deployment unit; and

all interaction between a process component in one deployment unit and any other process component in any other deployment unit takes place through the respective service operations of the two process components.

13. The product of claim 12, wherein the deployment units comprise:

a catalog authoring deployment unit that includes the product catalog authoring process component; and

a catalog publishing deployment unit that includes the product catalog publishing process component.

14. The product of any one of claims 9 to 13, further comprising:

a catalog authoring deployment unit that implements the maintain catalog, the change transmission status, the change publication status, the change catalog based on publication cancellation, the change catalog based on item lock status, the change catalog based on content change publication status, the request catalog publication, the request catalog publication cancellation, the request catalog item lock, the request catalog publication content change, and the notify of catalog update service operations; and

a catalog publishing deployment unit that implements the maintain published product catalog, the cancel catalog publication, the lock published catalog items, the change published catalog content, the notify of publication transmission package check, the confirm catalog publication, the confirm catalog publication cancellation, the confirm catalog item lock, and the confirm catalog publication content change service operations.

15. The product of any one of claims 1U to 14, wherein:

the product catalog authoring process component includes a product catalog master data object, a product catalog update run business object, a product catalog file upload run business object, a product catalog duplication run business object, a product catalog change list business object, a product catalog update method business object, and a product catalog publishing sending run business object, wherein:

the product catalog master data object is the business object responsible for the structured directory of catalog items where each catalog item represents a product and provides information about it;

the product catalog update run business object is the business object responsible for the run to update product catalogs based on product catalog update methods;

the product catalog file upload run business object is the business object responsible for the run to update a product catalog based on an uploaded file;

the product catalog duplication run business object is the business object responsible for the run to copy a product catalog in total or in parts;

the product catalog change list business object is the business object responsible for the list of changes to a product catalog;

the product catalog update method master data object is the business object responsible for the set of rules controlling how products are cataloged in a product catalog and how to update the product catalog when cataloged products are changed; and

the product catalog publishing sending run is the business object responsible for the run to trigger sending a product catalog for publication; and

the product catalog publishing process component includes a published product catalog master data object and a published product catalog update run business object, wherein:

the published product catalog master data object is the business object responsible for the version of a product catalog that has been released for access by or exchange with the target group of people for whom the product catalog has been tailored; and

the published product catalog update run business object is the business object responsible for the run to update a product catalog based on an uploaded file.

16. The product of any one of claims 10 to 15, wherein:

each of the process components includes one or more business objects; and

none of the business objects of any one of the process components interacts directly with any of the business objects included in any of the other process components.

17. The product of claim 16, wherein the business objects comprise a business process object.

18. The product of claim 16 or 17, wherein:

none of the business objects included in any one of the process components is included in any of the other process components.

19. The product of any one of claims 9 to 18, further comprising:

a plurality of process agents, each process agent being either an inbound process agent or an outbound process agent, an inbound process agent being operable to receive a message from an inbound operation, an outbound process agent being operable to cause an outbound operation to send a message, each process agent being associated with exactly one process component;

wherein:

the inbound process agents comprise a first inbound process agent operable to start the execution of the step requested in a first inbound message by creating or updating one or more business object instances; and

the outbound process agents comprise a first asynchronous outbound process agent that is called after a business object that is associated with the first outbound process agent changes.

20. A method for providing services from a computer system having product catalog functionality, the method comprising:

providing services through the sending of messages, the messages including:

a notification to update a product catalog;

a notification to change the transmission status of a product catalog;

a notification to change the publication status of a product catalog;

a notification to update a product catalog based on a publication cancellation;

a notification to update a product catalog based on a locked item;

a notification to update a product catalog based on a catalog change content publication status;

a notification requesting that a product catalog be published;

a notification requesting a product catalog publication be cancelled;

a notification requesting an item in a product catalog to be locked;

a notification requesting a content change in a product catalog publication;

a notification about an update to a product catalog;

a notification to update a published product catalog;

a notification to cancel the publication of a product catalog;

a notification to lock items contained in a published product catalog;

a notification to change the content in a published product catalog;

a notification about a catalog publication transmission package check;

a notification to confirm the publication of a product catalog;

a notification to confirm the cancellation of a product catalog publication;

a notification to confirm the lock of an item in a product catalog; and

a notification to confirm the content change of a product catalog publication.

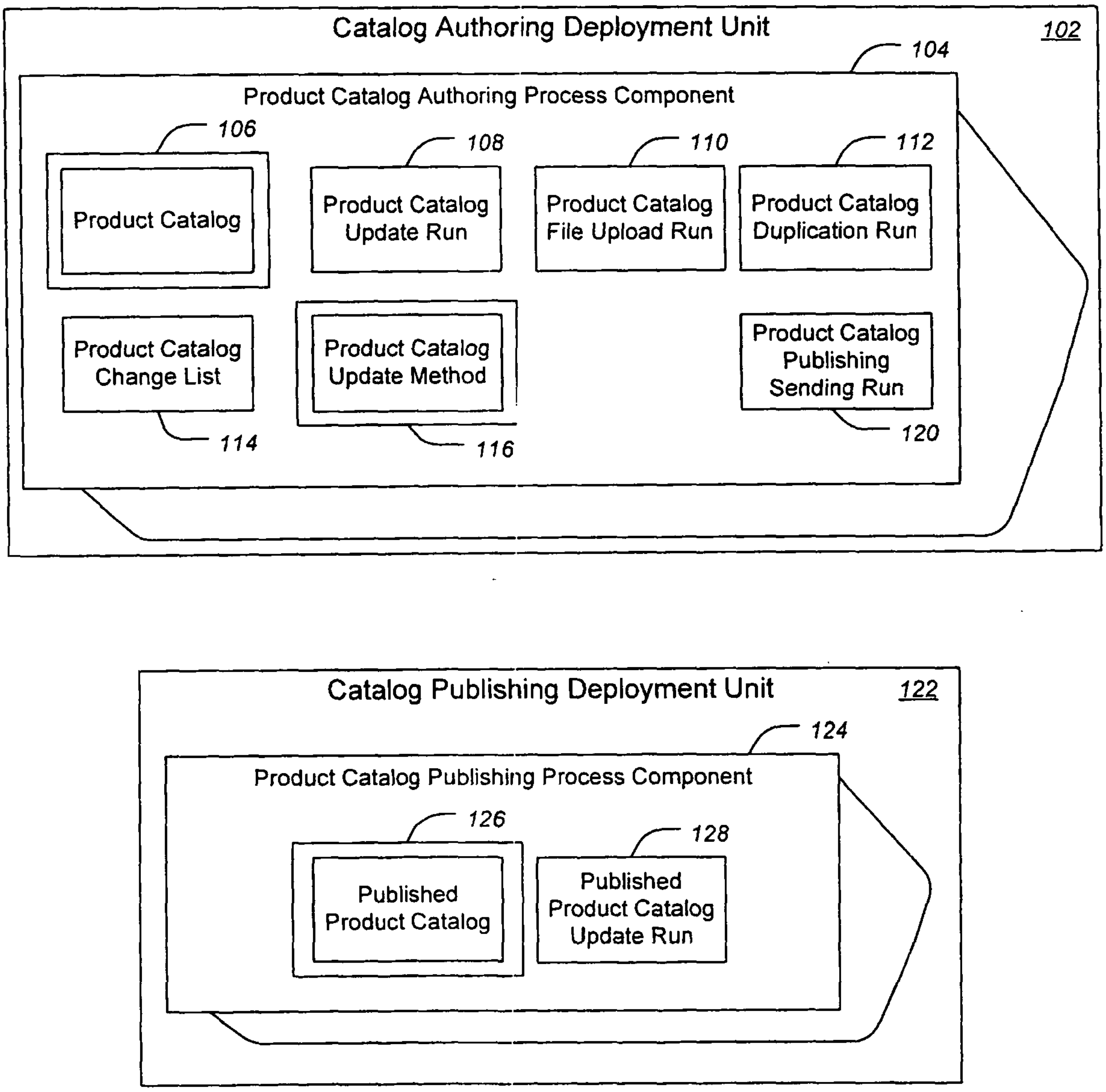
Catalog Authoring Deployment Unit
102
104
Product Catalog Authoring Process Component
106
Product Catalog
108
Product Catalog Update Run
110
Product Catalog File Upload Run
112
Product Catalog Duplication Run
Product Catalog Change List
114
Product Catalog Update Method
116
Product Catalog Publishing Sending Run
120
Catalog Publishing Deployment Unit
122
124
Product Catalog Publishing Process Component
126
Published Product Catalog
128
Published Product Catalog Update Run

FIG. 1

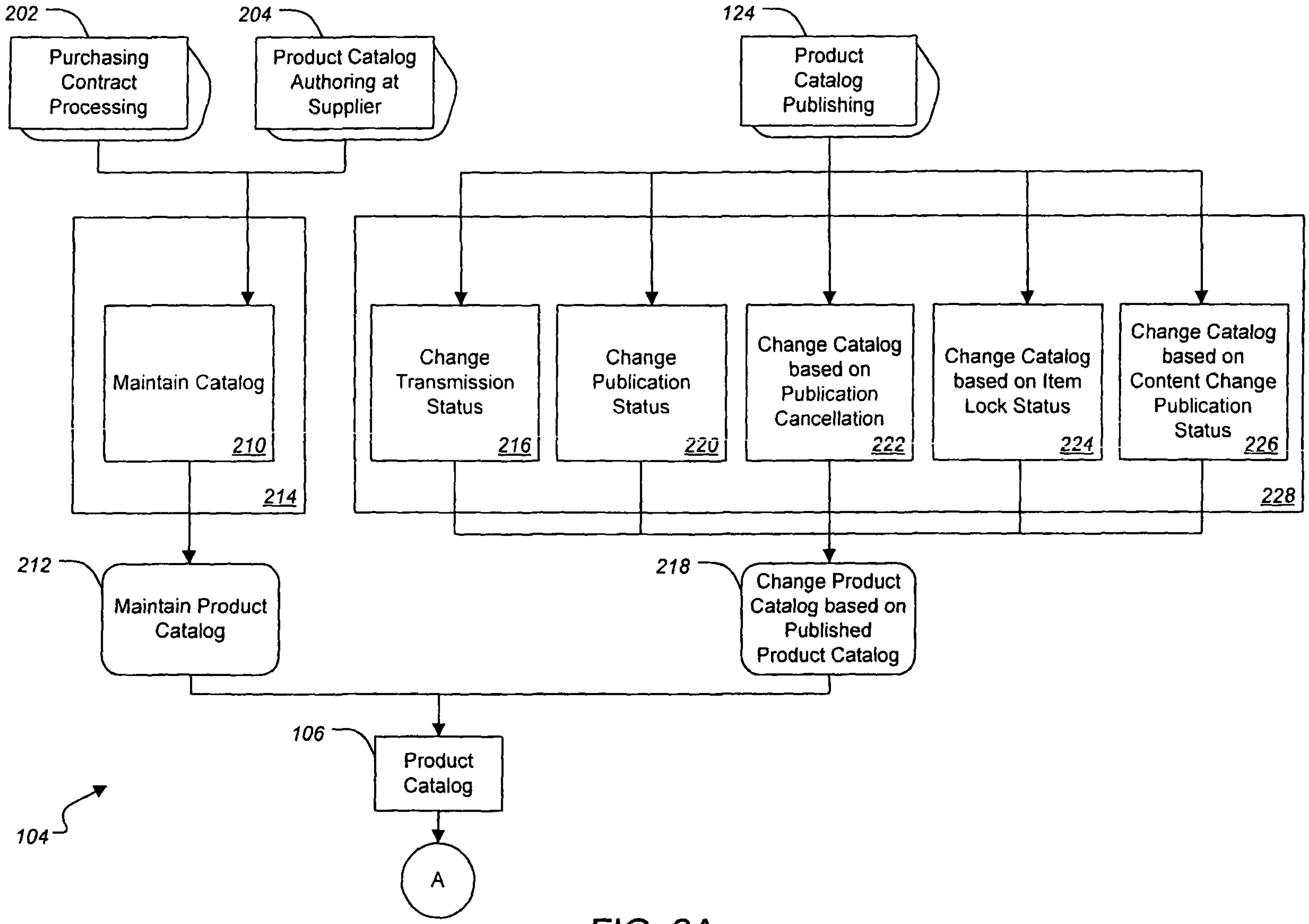
202
Purchasing Contract Processing
204
Product Catalog Authoring at Supplier
124
Product Catalog Publishing
Maintain Catalog
210
214
Change Transmission Status
216
Change Publication Status
220
Change Catalog based on Publication Cancellation
222
Change Catalog based on Item Lock Status
224
Change Catalog based on Content Change Publication Status
226
228
212
Maintain Product Catalog
218
Change Product Catalog based on Published Product Catalog
106
Product Catalog
A
104

FIG. 2A

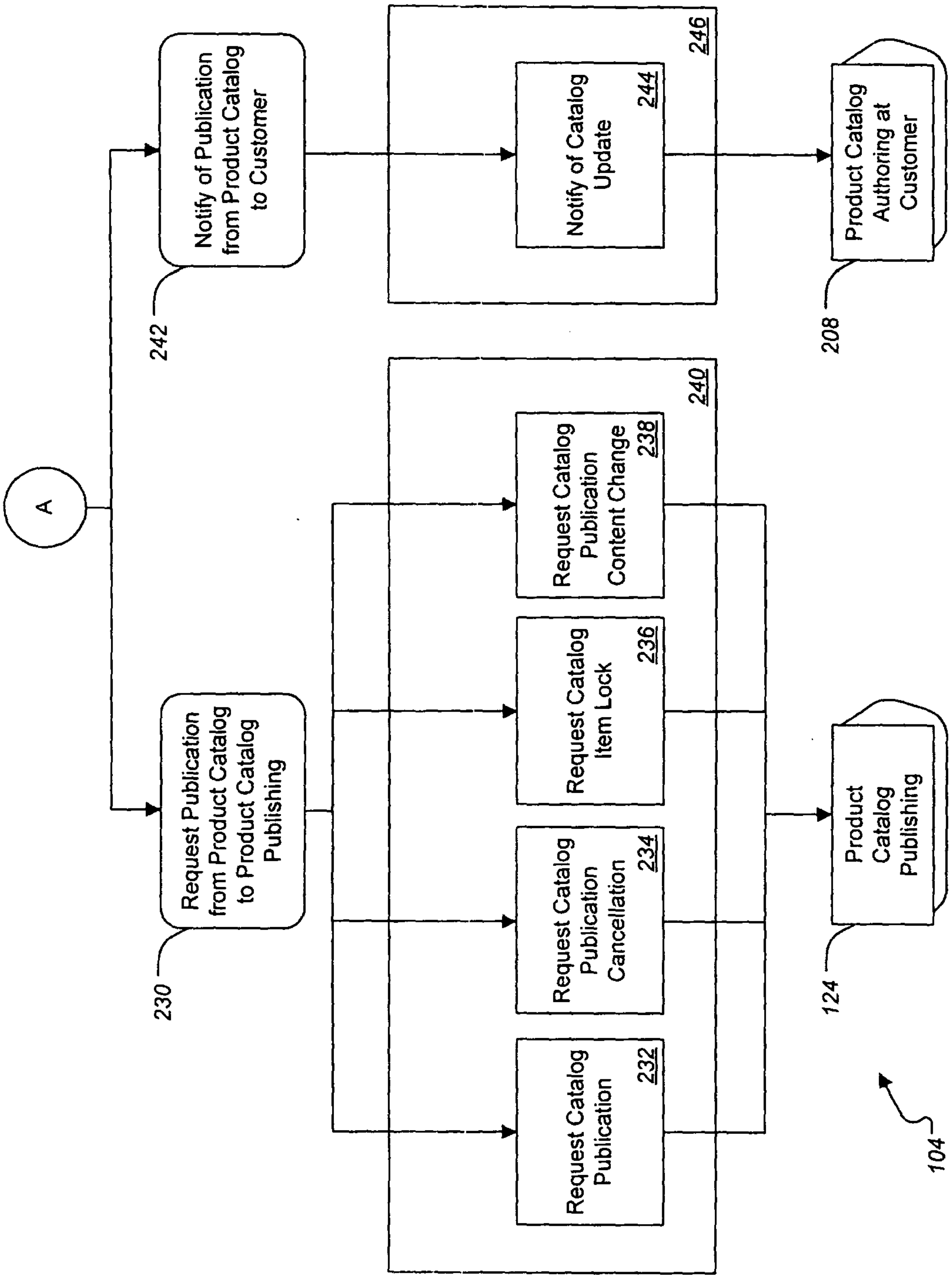
A
Request Publication from Product Catalog to Product Catalog Publishing
230
Notify of Publication from Product Catalog to Customer
242
Request Catalog Publication
232
Request Catalog Publication Cancellation
234
Request Catalog Item Lock
236
Request Catalog Publication Content Change
238
240
Notify of Catalog Update
244
246
Product Catalog Publishing
124
Product Catalog Authoring at Customer
208
104

FIG. 2B

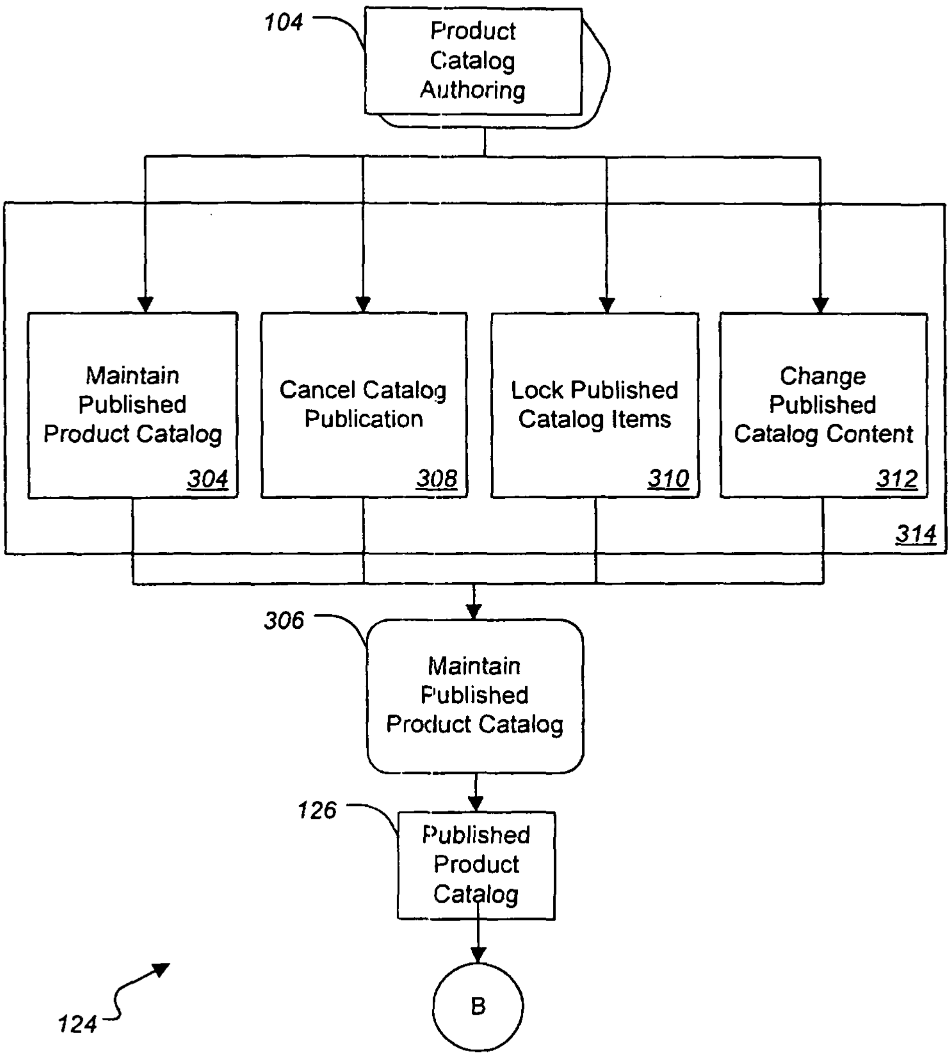
104
Product Catalog Authoring
Maintain Published Product Catalog
304
Cancel Catalog Publication
308
Lock Published Catalog Items
310
Change Published Catalog Content
312
314
306
Maintain Published Product Catalog
126
Published Product Catalog
B
124

FIG. 3A

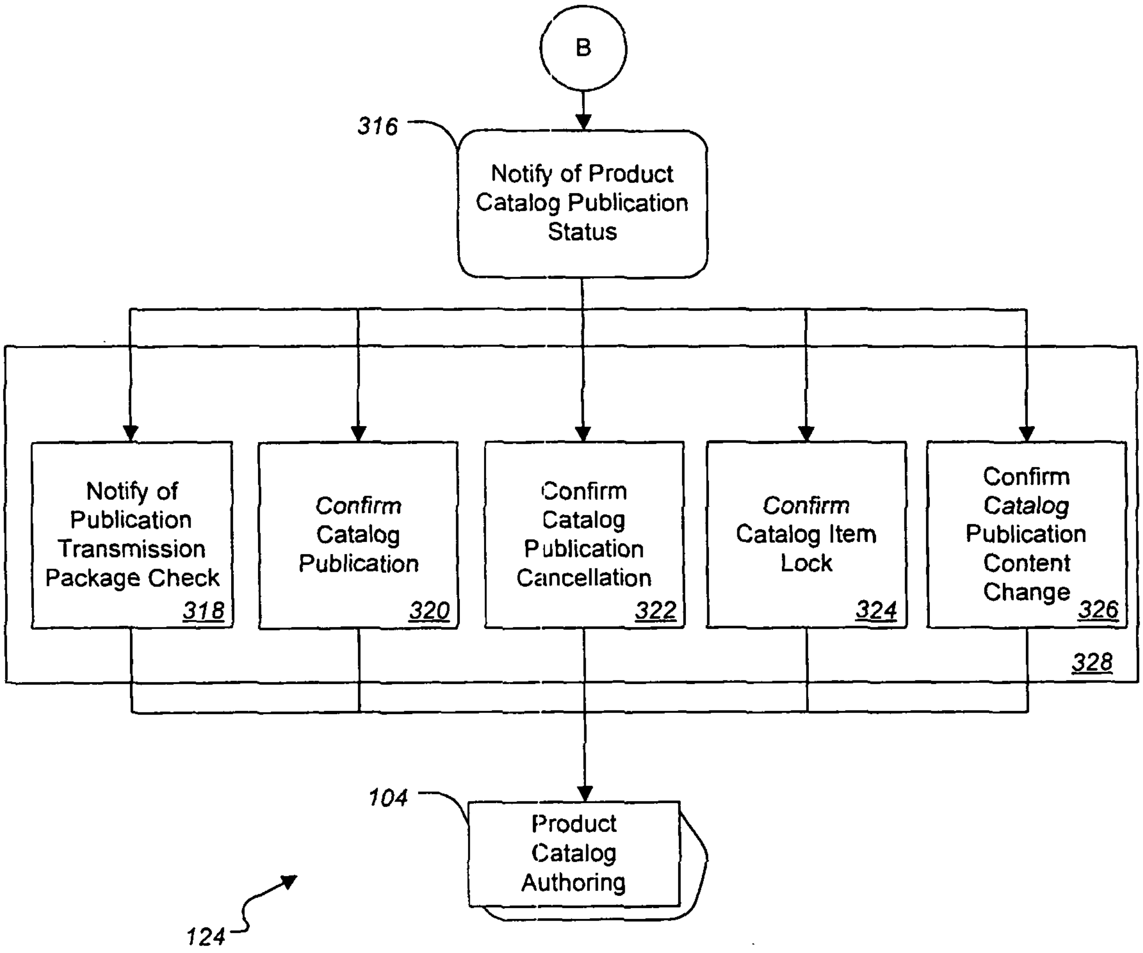
B
316
Notify of Product Catalog Publication Status
Notify of Publication Transmission Package Check
318
Confirm Catalog Publication
320
Confirm Catalog Publication Cancellation
322
Confirm Catalog Item Lock
324
Confirm Catalog Publication Content Change
326
328
104
Product Catalog Authoring
124

FIG. 3B

PATENT COOPERATION TREATY

PCT

DECLARATION OF NON-ESTABLISHMENT OF INTERNATIONAL SEARCH REPORT

(PCT Article 17(2)(a), Rules 13*ter*.1(c) and Rule 39)

Applicant's or agent's file reference S 8610WO-ro/za	IMPORTANT DECLARATION	Date of mailing *(day/month/year)* 17/07/2007
International application No. PCT/EP2007/002842	International filing date *(day/month/year)* 29/03/2007	(Earliest) Priority date *(day/month/year)* 30/03/2006
International Patent Classification (IPC) or both national classification and IPC G06Q30/00		
Applicant SAP AG		

This International Searching Authority hereby declares, according to Article 17(2)(a), that **no international search report will be established** on the international application for the reasons indicated below

1. ☒ The subject matter of the international application relates to:
   - a. □ scientific theories
   - b. □ mathematical theories
   - c. □ plant varieties
   - d. □ animal varieties
   - e. □ essentially biological processes for the production of plants and animals, other than microbiological processes and the products of such processes
   - f. ☒ schemes, rules or methods of doing business
   - g. □ schemes, rules or methods of performing purely mental acts
   - h. □ schemes, rules or methods of playing games
   - i. □ methods for treatment of the human body by surgery or therapy
   - j. □ methods for treatment of the animal body by surgery or therapy
   - k. □ diagnostic methods practised on the human or animal body
   - l. □ mere presentations of information
   - m. □ computer programs for which this International Searching Authority is not equipped to search prior art
2. ☒ The failure of the following parts of the international application to comply with prescribed requirements prevents a meaningful search from being carried out:
   □ the description ☒ *the claims* □ *the drawings*
3. □ A meaningful search could not be carried out without the sequence listing; the applicant did not, within the prescribed time limit:
   - □ furnish a sequence listing on paper complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
   - □ furnish a sequence listing in electronic form complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
   - □ pay the required late furnishing fee for the furnishing of a sequence listing in response to an invitation under Rule 13*ter*.1(a) or (b).
4. □ A meaningful search could not be carried out without the tables related to the sequence listings; the applicant did not, within the prescribed time limit, furnish such tables in electronic form complying with the technical requirements provided for in Annex C-*bis* of the Administrative Instructions, and such tables were not available to the International Searching Authority in a form and manner acceptable to it.
5. Further comments:

Name and mailing address of the International Searching Authority 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* Roger Thomas

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 203

The claims relate to subject matter for which no search is required according to Rule 39.1 PCT (Scheme, rules and method for doing business). Given that the claims are formulated in terms of such subject matter or merely specify commonplace features relating to its technological implementation, the search examiner could not establish any technical problem which might potentially have required an inventive step to overcome. Hence it was not possible to carry out a meaningful search into the state of the art (Art. 17(2)(a)(i) and (ii) PCT; see PCT International Search Guidelines, Chapter 9).

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried out during examination before the EPO (see EPO Guideline C-VI, 8.5), should the problems which led to the Article 17(2) declaration be overcome.