ENTERPRISE COMPUTING WITH OBJECTS pdf

1: In Enterprise Computing, what are the differences between objects and components? | Yahoo Answers

Includes bibliographical references (pages) and index Enterprise Computing with Objects helps the reader place rapidly evolving technologies - the Internet, the World Wide Web, distributed computing, object technology and client/server systems - in their appropriate context in relation to modern information systems.

The individual data objects are used in addition to business partner master data, and also in some cases in addition to generic product master data. The individual data objects are configurable such that attributes for an individual data object change over the lifetime of the unique asset to which the individual data object pertains. Techniques for modeling the lifecycle of an individual object family are described. Techniques also are provided for an integration framework that integrates a software application with an individual object that is used or altered by the software application. TECHNICAL FIELD This invention relates to computing master data that are structured as individual data objects wherein each such object uniquely defines a single asset, and also relates to integrating business computing processes with the individual data objects. BACKGROUND Customer relationship management CRM systems are designed to track relationships between a business and its customers which may be referred to as a type of business partner , as well as the services or assets such as software, contracts, or property that form these relationships. Assets are typically tracked through records representing business partners that have relationships to certain assets. This is the case because CRM systems are designed to focus primarily on customers, or business partners. Additionally, records of these assets are sometimes kept in a tabular format, with a single unique identifier. Maintenance of data consistency between different aspects of a comprehensive CRM system such as aspects of service, marketing, and sales that relate to the same services or assets can become increasingly complex as additional systems are used for transactions with the same service or asset, especially where different systems use different records to represent the same particular service or asset. Transactions and other events in CRM systems are typically initiated by determining the type of transaction desired. Parameters such as customer, cost and time are set, and then a suitable service or asset of the transaction must be found and selected, after which the appropriate modifications must be made to all of the various records that can represent the service or asset. A shortcoming of CRM and other similar systems is that tracking individual assets can become difficult in industries where assets are high-valued and have long life-spans. In these areas, assets often become increasingly unique with the passage of time. Some CRM systems use product master data for representing assets, wherein a single generic record of master data is used to list all of the components and characteristics of all assets of a particular type. Existing systems are also oriented toward relationships with customers or other business partners, and not towards relationships with assets. Thus, in some cases, it is awkward for such systems to accurately reflect individual assets as they become more and more unique. SUMMARY In one aspect, the invention provides for individual data objects in an enterprise computing system, where the individual data objects represent unique assets of a family of assets. Thus, the enterprise computing system enables the modeling of business processes centrally around uniquely defined individual objects, while still retaining the advantages of a customer, or business partner, focused system. Such modeling is particularly useful for certain types of assets, such as assets that are highly configurable, have a high value, or have a long and complex lifecycle. In another aspect, the invention provides a computer system for processing individual data objects. The system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family, and includes, when first created, a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object based on user input, and that associates an attribute value with each added attribute. In various implementations, the computer system has one or more of the following features. The object family may identify the attributes that are permitted to be added to each individual object associated with the object family, and the executable software module may only add an attribute to an individual object when the attribute is identified by the object family as a permitted attribute. Each individual object in the data repository may include an attribute value for

each attribute of the individual object. The asset that an individual data object represents may be, for example, a tangible or intangible product, or a service. The computer system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family and includes, when first created, only a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and that associates an attribute value with each added attribute. In various implementations, the object family may identify the attributes that are permitted to be added to each individual object associated with the object family, and the executable software module only adds an attribute to an individual object when the attribute is identified by the object family as a permitted attribute. In another aspect, the invention provides a data repository including individual objects. Each individual object in the repository is associated with an object family and has a unique identifier attribute and attributes that are common with other individual objects that are associated with the same object family to which the object is associated. Each such attribute has a corresponding attribute value. In various implementations, each individual object in the repository pertains to a unique asset within an asset family. Additional attributes may be added to each individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and an attribute value may be associated with each added attribute. The repository may include asset information, information for a product, information for a service, and information for an intangible product. The repository may also include business partner information, with each individual object being capable of being associated with a business partner. In another aspect, the invention provides a data repository with product master data, business partner master data, and individual objects. The product master data includes product master records. Each product master record is associated with a particular type of product. The business partner master data includes business partner master records. Each business partner master record is associated with a particular business partner. As for the individual objects, each individual object a pertains to a unique asset within an asset family, b is associated with an object family corresponding to the asset family, c includes, when first created, a unique identifier and attributes, each attribute being associated with an attribute value, and d is associated with a particular product master record and a particular business partner master record. Additional attributes are added to an individual object in response to a temporal event having impact on the asset to which the individual object pertains, and attribute values are associated with each added attribute. In another aspect, the invention provides an integration system to integrate a computer system executing a software application with uniquely defined individual data objects. Each individual data object has firstly, attributes, secondly, allowable states for the attributes, and thirdly, allowable transitions between the allowable states. The integration system has an executable software module, wherein, in response to the integration system receiving, from the computer system executing the software application, information that requires an action to be performed on an identified one of the uniquely defined individual data objects, the executable software module produces a corresponding triggering event. The integration system also has data containers. The data containers are, firstly, for importing context data for the raised event, and secondly, for exporting context data returned by the identified individual data object in response to the generated triggering event. The integration system further has an executable event manager software module. This module, firstly, determines, based on the raised triggering event and any imported context data, an action to be performed on the identified individual data object, the event manager using a state manager to determine any transition in the allowable state of the identified individual data object that is required to be made in response to the generated triggering event. Secondly, the event manager software module executes the action, including any required state change, on the identified individual data object. In various implementations, the individual data objects may represent a configurable item whose configuration changes over the lifecycle of the item. The item may be a product such as an automobile, or a service, for example. The information received from the computer system may explicitly identify the triggering event, or it may do so implicitly. The individual object attributes may include an identifier that uniquely identifies an item corresponding to the particular uniquely defined individual object. The unique identifier may be, for example,

a vehicle identification number for an automobile. Possible triggering events include a data request, for example. The integration may further comprise a state manager adapter that interfaces the event manager with the state manager, which may be part of the integration system or alternatively may be in an external system called by the integration system. The state manager may include a state machine engine, and may include a repository of allowable states for individual data objects and a repository of allowable transitions between states. Upon execution of the action on the individual object, a data container may be populated with context data from the identified individual object. The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims. Like reference symbols in the various drawings indicate like elements. The group of systems includes an enterprise computing system , which is networked with a supplier network on a buy-side , and also with a customer network on a sell-side. In this example, the enterprise system includes a supply chain management SCM solution that links purchasing-related processes with the business procedures of suppliers. A customer relationship management CRM solution guarantees sales-related links with customers, and a supplier relationship management SRM solution guarantees supply-related links with suppliers. The association between the two external systems and is enabled by a common enterprise resource planning ERP system In the enterprise system , master data is stored centrally in a data repository The master data, in this example, includes individual data object information, business partner information, and product information. Master data tools are used, for example, to create and maintain the master data. Configuration tools may also be used to structure the master data. A wide array of business applications make use of the centrally stored master data. An individual data object, or simply individual object, represents a globally unique tangible or intangible product. By way of example, individual objects represent the products that a company sells, or for which the company provides a service. The following are examples of assets that an individual data object may represent: Examples of object, or asset, families include, for example, household appliances, software licenses, and so on. Establishing an object family for an individual object may establish which fields, or attributes, are maintained with the individual object. The data structure of an individual object may have a very small kernel. The kernel includes, for example, a unique identifier for the individual object and, in some cases, attributes stored as part of the individual object. Data fields or attributes and groups of fields which are needed by the individual objects may be created, and then those fields and groups of fields may be linked to the kernel. As such, the individual object need not have a fixed set of fields that may never be used, and thus would waste data storage. The attributes of an individual object may include information about the asset corresponding to the individual object. Examples of such attributes include technical data, configuration information, counters, wage information, and documents. An individual object may also have relations to business partners persons or organizations. These relationships may be time dependent. For example, a person with an ownership relationship to an individual object may not have that relationship over the entire lifecycle of the individual object. An individual object may also have attribute information identifying a location of the asset corresponding to the object that is, geographical data. Examples of location data include a site where a piece of equipment the unique asset is installed, a geographical location where a mobile asset e. The individual object may also have an object calendar with dates and tasks. The use of individual object master data is especially useful in industries where products are high valued or have a long life span, in which cases each product may become increasingly unique throughout its lifecycle. In addition, some types of products are becoming more and more configurable, resulting in an increase in product options, and thus unique products. In these scenarios, the use of individual object data is especially useful in that a central reference for transactions between a company and its business partners may be the individual object. It is possible, as is described in more detail later, to perform business transactions on the individual objects. In a CRM system, for example, this may include various sales, marketing and service transactions. For example, individual objects may be embedded in sales processes, service processes, marketing processes, or in processes for getting information from customer interaction, or call center, applications. In another example, individual objects may be used in marketing applications that determine a target group of customers or potential customers toward

which to direct a marketing campaign.

ENTERPRISE COMPUTING WITH OBJECTS pdf

2: USA1 - Individual data objects in enterprise computing systems - Google Patents

Enterprise Computing with Objects helps the reader place rapidly evolving technologies-the Internet, the World Wide Web, distributed computing, object technology and client/server systems-in their appropriate context in relation to modern information systems.

The individual data objects are used in addition to business partner master data, and also in some cases in addition to generic product master data. The individual data objects are configurable such that attributes for an individual data object change over the lifetime of the unique asset to which the individual data object pertains. Techniques for modeling the lifecycle of an individual object family are described. Techniques also are provided for an integration framework that integrates a software application with an individual object that is used or altered by the software application. Assets are typically tracked through records representing business partners that have relationships to certain assets. This is the case because CRM systems are designed to focus primarily on customers, or business partners. Additionally, records of these assets are sometimes kept in a tabular format, with a single unique identifier. Transactions and other events in CRM systems are typically initiated by determining the type of transaction desired. Parameters such as customer, cost and time are set, and then a suitable service or asset of the transaction must be found and selected, after which the appropriate modifications must be made to all of the various records that can represent the service or asset. In these areas, assets often become increasingly unique with the passage of time. Some CRM systems use product master data for representing assets, wherein a single generic record of master data is used to list all of the components and characteristics of all assets of a particular type. Existing systems are also oriented toward relationships with customers or other business partners, and not towards relationships with assets. Thus, in some cases, it is awkward for such systems to accurately reflect individual assets as they become more and more unique. Thus, the enterprise computing system enables the modeling of business processes centrally around uniquely defined individual objects, while still retaining the advantages of a customer, or business partner, focused system. Such modeling is particularly useful for certain types of assets, such as assets that are highly configurable, have a high value, or have a long and complex lifecycle. The system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family, and includes, when first created, a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object based on user input, and that associates an attribute value with each added attribute. The object family may identify the attributes that are permitted to be added to each individual object associated with the object family, and the executable software module may only add an attribute to an individual object when the attribute is identified by the object family as a permitted attribute. Each individual object in the data repository may include an attribute value for each attribute of the individual object. The asset that an individual data object represents may be, for example, a tangible or intangible product, or a service. The computer system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family and includes, when first created, only a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and that associates an attribute value with each added attribute. Each individual object in the repository is associated with an object family and has a unique identifier attribute and attributes that are common with other individual objects that are associated with the same object family to which the object is associated. Each such attribute has a corresponding attribute value. Additional attributes may be added to each individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and an attribute value may be associated with each added attribute. The repository may include asset information, information for a product, information for a service, and information for an intangible product. The repository may also include business partner

information, with each individual object being capable of being associated with a business partner. The product master data includes product master records. Each product master record is associated with a particular type of product. The business partner master data includes business partner master records. Each business partner master record is associated with a particular business partner. As for the individual objects, each individual object a pertains to a unique asset within an asset family, b is associated with an object family corresponding to the asset family, c includes, when first created, a unique identifier and attributes, each attribute being associated with an attribute value, and d is associated with a particular product master record and a particular business partner master record. Additional attributes are added to an individual object in response to a temporal event having impact on the asset to which the individual object pertains, and attribute values are associated with each added attribute. Each individual data object has firstly, attributes, secondly, allowable states for the attributes, and thirdly, allowable transitions between the allowable states. The integration system has an executable software module, wherein, in response to the integration system receiving, from the computer system executing the software application, information that requires an action to be performed on an identified one of the uniquely defined individual data objects, the executable software module produces a corresponding triggering event. The integration system also has data containers. The data containers are, firstly, for importing context data for the raised event, and secondly, for exporting context data returned by the identified individual data object in response to the generated triggering event. The integration system further has an executable event manager software module. This module, firstly, determines, based on the raised triggering event and any imported context data, an action to be performed on the identified individual data object, the event manager using a state manager to determine any transition in the allowable state of the identified individual data object that is required to be made in response to the generated triggering event. Secondly, the event manager software module executes the action, including any required state change, on the identified individual data object. The item may be a product such as an automobile, or a service, for example. The information received from the computer system may explicitly identify the triggering event, or it may do so implicitly. The individual object attributes may include an identifier that uniquely identifies an item corresponding to the particular uniquely defined individual object. The unique identifier may be, for example, a vehicle identification number for an automobile. Possible triggering events include a data request, for example. The integration may further comprise a state manager adapter that interfaces the event manager with the state manager, which may be part of the integration system or alternatively may be in an external system called by the integration system. The state manager may include a state machine engine, and may include a repository of allowable states for individual data objects and a repository of allowable transitions between states. Upon execution of the action on the individual object, a data container may be populated with context data from the identified individual object. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims. The group of systems includes an enterprise computing system , which is networked with a supplier network on a buy-side , and also with a customer network on a sell-side. In this example, the enterprise system [] includes a supply chain management SCM solution that links purchasing-related processes with the business procedures of suppliers. A customer relationship management CRM solution guarantees sales-related links with customers, and a supplier relationship management SRM solution guarantees supply-related links with suppliers. The association between the two external systems and is enabled by a common enterprise resource planning ERP system In the enterprise system [] , master data is stored centrally in a data repository The master data, in this example, includes individual data object information, business partner information, and product information. Master data tools are used, for example, to create and maintain the master data. Configuration tools may also be used to structure the master data. A wide array of business applications make use of the centrally stored master data. An individual data object, or simply individual object, represents a globally unique tangible or intangible product. By way of example, individual objects represent the products that a company sells, or for which the company provides a service. The following are examples of assets that an individual data object may represent: Examples of object, or asset, families include, for example, household appliances, software licenses, and so on. Establishing an object family for an individual object may establish which fields, or attributes, are

maintained with the individual object. The kernel includes, for example, a unique identifier for the individual object and, in some cases, attributes stored as part of the individual object. Data fields or attributes and groups of fields which are needed by the individual objects may be created, and then those fields and groups of fields may be linked to the kernel. As such, the individual object need not have a fixed set of fields that may never be used, and thus would waste data storage. Examples of such attributes include technical data, configuration information, counters, wage information, and documents. An individual object may also have relations to business partners persons or organizations. These relationships may be time dependent. For example, a person with an ownership relationship to an individual object may not have that relationship over the entire lifecycle of the individual object. An individual object may also have attribute information identifying a location of the asset corresponding to the object that is, geographical data. Examples of location data include a site where a piece of equipment the unique asset is installed, a geographical location where a mobile asset e. The individual object may also have an object calendar with dates and tasks. In addition, some types of products are becoming more and more configurable, resulting in an increase in product options, and thus unique products. In these scenarios, the use of individual object data is especially useful in that a central reference for transactions between a company and its business partners may be the individual object. It is possible, as is described in more detail later, to perform business transactions on the individual objects. In a CRM system, for example, this may include various sales, marketing and service transactions. For example, individual objects may be embedded in sales processes, service processes, marketing processes, or in processes for getting information from customer interaction, or call center, applications. In another example, individual objects may be used in marketing applications that determine a target group of customers or potential customers toward which to direct a marketing campaign. In this example, the customers may be identified by virtue of a relationship they have to certain individual objects, or that they have to certain individual objects with certain attributes. By way of example, a business partner may be a supplier, a customer or potential customer , an employee of the enterprise, a supplier or a customer , or any other person or entity. The information about a business partner included in the business partner master data may include, for example, address information, contact persons, relationships between different business partners, and credit, payment and delivery information. In comparison to individual object master data, product master data relates to a product, whereas individual object master data relates to a specific unique one of the product. By way of example, a product may be a specific model of refrigerator, and an individual object may be a specific refrigerator with a unique serial number. The product master data products can be tangible such as a personal computer or intangible such as services like personal computer maintenance. Product-specific data may be stored in specific product master records. Each product master record ideally contains the data required to manage the product. This data can be divided into 1 data of a descriptive nature such as size and weight; and 2 data with a control function, such as an item category group for determining each item in a purchase order. In this example, the individual object is a wheel loader, which is a construction machine. The loader is modeled as an individual object, as indicated by the circle Individual objects, such as the object shown in FIG. All three of these attribute types are shown in the example shown in FIG. Also, there may be a history of attribute values that may be archived for each of these attributes in a history record for the individual object. The component list may be modeled, for example, by a configuration management tool or an installed base tool. The modeled component list object may then be linked to the individual object, as depicted in FIG. In this example, the main components of the loader are an engine, a driving cab, and a crane. The engine itself is an individual object, and the driving cab and crane are products because, for example, these components may not be tracked individually as individual objects.

ENTERPRISE COMPUTING WITH OBJECTS pdf

3: USB2 - Individual data objects in enterprise computing systems - Google Patents

Enterprise Computing with Objects concludes with some forecasting about the (predictably bright) future of objects in distributed systems. Overall, this is a very useful book, which presents a lot of critical information about wide-ranging roles for objects, now and in the future.

In our view, one key to the success of this evolution is object technology. With its modular approach to development, object technology addresses many of the key issues of enterprise computing. It is good at managing complexity, making software components available for reuse, and managing changes, such as the rapid evolution of the World Wide Web. In the process, it transformed the face of human-computer interaction, giving us graphical user interfaces GUIs and object-oriented user interfaces OOUIs. The World Wide Web has introduced a new and pervasive user interface platform and object technology has expanded out of its client-only role. We are facing a transition with a profound impact on existing information systems, as well as on systems yet to be constructed. Technological advancement always produces confusion, risks, and opportunities. There is great value in understanding the essential elements of the technology as opposed to the incidental ones. Notice that we say the value is in understanding rather than knowing. People can know dozens of buzzwords and hundreds of implementation details, but if they do not have a fundamental understanding of how and why the pieces fit together, their knowledge will soon become obsolete. This book is an architectural overview and a digest of essential topics, covering a broad and complex subject as quickly as possible. We separate what is essential from what is incidental so that as products continue to evolve, the book will continue to have value. Such a book could easily run over a thousand pages. However, because your time is precious, this book distills and filters the information so that a busy professional can receive the maximum benefit in the shortest amount of time. We do not attempt to provide comprehensive evaluations or recommendations of products. Although we include many details of products and systems, we use these to illustrate our concepts and ensure linkage between our concepts and the real world. Because we cannot cover everything in detail, we include comprehensive references that point the reader to more detailed information sources. What Readers Can Hope to Gain En route to understanding enterprise computing with objects, readers will gain a clear picture of different types of clients, servers, and the software "glue" that holds systems together. They will also come to understand why object technology was introduced to enterprise computing and where the combination is heading. They will learn how objects can be introduced into various implementations, as well as the benefits and drawbacks of each case. After reading the book, readers should be able to better see through the marketing hype surrounding both object technology and enterprise computing, and they should be able to focus on the essential architecture issues relevant to solving real-life business and computing problems. The World Wide Web is, of course, the fastest growing area of computing, and object technology has profound implications for the Web. Who This Book Is For The book is aimed primarily at technical professionals, such as application developers, systems analysts, managers, and system architects. However, other technical professionals can benefit from the book, as can students of computer science, management information systems, and computer engineering. Nontechnical managers can benefit from our general overview of concepts and concerns that affect their computing systems. This kind of technical understanding can be valuable in helping people reach their career goals. We assume that readers do not necessarily have prior application programming experience. Although the book is potentially valuable for programmers, it is not a programming manual or a "hands-on" guide. Rather, it can help programmers gain a high-level architectural understanding of areas they may be less familiar with. Once this groundwork is established, we look at a series of possible approaches for incorporating objects into enterprise computing systems. After looking at object persistence, we examine the ways enterprises can leverage their existing knowledge base to move their information system onto the Internet and the World Wide Web. We follow this with an examination of the development tools that make end-to-end enterprise and Internet development possible. The concluding chapter offers a summary and a direction for the future. Using this Book Although this book is meant to be read from cover to cover, it can also be used in less structured ways. Summaries at the end of each

chapter offer a digest of the essentials for easy reference. The glossary provides a comprehensive list of the terms used in this book. Sidebars provide more in-depth looks at related topics. Readers interested in specific topics can navigate by using the table of contents and the index. The comprehensive list of references provides a jumping-off spot for further research and study. Conventions For ease of reference, the book uses a two-column format. The sentences in the left column summarize the key points in the text. These summaries are not intended to be read sequentially. Instead, readers can use them to quickly review what they have read or to skim to find relevant information. Also, by quickly scanning the summary sentences above and below a section, a reader who jumps into a section can quickly establish the context for reading. Sidebars Sometimes a topic strikes us as interesting or relevant and needs further treatment, but it does not seem to fit into the immediate discussion. We have included these topics in sidebars like this one. Sidebars tend to cover related technological issues. They can also cover our own opinions or speculations on a subject. The book contains a large number of figures to illustrate technical concepts, system configurations, and other items of interest. We use the following conventions to denote the different components of enterprise systems. Database Transaction processing monitor TP monitor with message queue Programs Traditional 3GL program written in a non-OO language, such as Assembler, C, and COBOL OO program, using connected circles to depict objects and their references to each other Executing Programs Executing programs are depicted as programs inside an executing environment, as in the following two examples: Acknowledgments Many people have helped us with this book. Although it is impossible to thank all of them, we will try. To our reviewers, for their numerous helpful and insightful comments: Mark Landry, Lloyd W. To Hayden Lindsey, whose early work and discussion inspired Chapter 9. To our IBM managers who encouraged us along the way: To many friends, mentors, and colleagues in and outside IBM from whom we have learned so much in numerous discussions and exchanges over the years. We list them in alphabetical order: To Tom Love, who encouraged Shan to create the book. Smith and Fred Brooks, who taught Shan, among many other things, the discipline of writing ten minutes every day, no matter how great the workload at the time. Without this discipline, the book, which took five years to finish, would not have been possible. Special thanks to our editor, Carter Shanklin, and to the other people at Addison Wesley Longman who made this book possible: To our families, who have been steadfast supporters throughout this long journey, for their invaluable love and understanding.

ENTERPRISE COMPUTING WITH OBJECTS pdf

4: Gartner Top 10 Strategic Technology Trends for - Smarter With Gartner

Description: Shan and Earle show how to merge client/server technology and object technology to build systems that offer real business advantages, and explain how object technology and enterprise computing relate to each other now and in the future.

What is claimed is: A computer system for processing individual data objects, the system comprising: The computer system of claim 1 wherein the second tangible asset represented by the second individual data object is a component included in the first tangible asset represented by the first individual data object. A data storage device comprising a data repository comprising: The data storage device comprising the data repository of claim 4 wherein each individual data object corresponds to a unique tangible asset within an asset family. The data storage device comprising the data repository of claim 5 wherein the executable software module is configured to add an attribute and an attribute value for an added attribute to an individual data object in response to a temporal event having an impact on the unique tangible asset to which the individual data object corresponds. The data storage device comprising the data repository of claim 5 wherein the repository includes asset information. The data storage device comprising the data repository of claim 7 wherein the asset information includes information for a product. The data storage device comprising the data repository of claim 4 wherein: The data storage device comprising the data repository of claim 4 wherein each individual data object is configured to include configuration information for the individual data object. The data storage device comprising the data repository of claim 4 wherein each individual data object is configured to be associated with history information for the individual data object providing information related to attributes added to and deleted from the individual data object by the executable software module. The data storage device comprising the data repository of claim 4 wherein each individual data object is configured to include location information for the geographical location of the individual data object. The data storage device comprising the data repository of claim 4 wherein each individual data object is configured to be associated with calendar information that includes dates and tasks, each task being associated with a particular date. The computer system of claim 1 wherein the executable software module is configured to: The computer system of claim 1 wherein the data repository further includes: The computer system of claim 15 wherein: The computer system of claim 16 wherein: The computer system of claim 16 wherein the first individual data object includes a component list identifying the second individual data object and the third individual data object. The computer system of claim 1 wherein the first individual data object includes a second attribute and a second attribute value including information identifying a product data object representing a third tangible asset to link the first individual data object to the product data object, the product data object being a type of data object different from an individual data object and the third tangible asset being different from the first tangible asset and the second tangible asset. The computer system of claim 19 wherein: The computer system of claim 20 wherein: The computer system of claim 20 wherein the first individual data object includes a component list identifying the second individual data object and the product data object. The computer system of claim 23 wherein the third individual data object includes less attributes than the first individual data object. The computer system of claim 24 wherein the executable software module is configured to add a new attribute to the third individual data object such that, after the executable software module adds the new attribute to the third individual data object, the third individual data object includes a same number attributes as the first individual data object. The computer system of claim 25 wherein the executable software module is configured to add a second new attribute to the third individual data object such that after the executable software module, adds the second new attribute to the third individual data object, the third individual data object includes more attributes than the first individual data object. The computer system of claim 24 wherein the executable software module is configured to delete an old attribute from the first individual data object such that, after the executable software module deletes the old attribute to the first individual data object, the third individual data object includes a same number attributes as the first individual data object. The computer system of claim 27 wherein the executable software module is configured to delete a second old attribute from

the first individual data object such that, after the executable software module deletes the second old attribute to the first individual data object, the third individual data object includes more attributes than the first individual data object. The computer system of claim 23 wherein the third individual data object includes a set of attributes that is different than a set of attributes included in the first individual data object. The computer system of claim 1 wherein individual data objects only include attributes for which attribute values exist. The computer system of claim 1 wherein individual data objects only include an individual object kernel representing essential elements when first created. The computer system of claim 21 wherein the individual object kernel only includes: The computer system of claim 1 wherein the first individual data object includes an asset association record that includes: The computer system of claim 1 wherein the executable software module is configured to prevent addition of attributes other than those included in the object family to which the identified individual data object is associated. TECHNICAL FIELD This invention relates to computing master data that are structured as individual data objects wherein each such object uniquely defines a single asset, and also relates to integrating business computing processes with the individual data objects. BACKGROUND Customer relationship management CRM systems are designed to track relationships between a business and its customers which may be referred to as a type of business partner , as well as the services or assets such as software, contracts, or property that form these relationships. Assets are typically tracked through records representing business partners that have relationships to certain assets. This is the case because CRM systems are designed to focus primarily on customers, or business partners. Additionally, records of these assets are sometimes kept in a tabular format, with a single unique identifier. Maintenance of data consistency between different aspects of a comprehensive CRM system such as aspects of service, marketing, and sales that relate to the same services or assets can become increasingly complex as additional systems are used for transactions with the same service or asset, especially where different systems use different records to represent the same particular service or asset. Transactions and other events in CRM systems are typically initiated by determining the type of transaction desired. Parameters such as customer, cost and time are set, and then a suitable service or asset of the transaction must be found and selected, after which the appropriate modifications must be made to all of the various records that can represent the service or asset. A shortcoming of CRM and other similar systems is that tracking individual assets can become difficult in industries where assets are high-valued and have long life-spans. In these areas, assets often become increasingly unique with the passage of time. Some CRM systems use product master data for representing assets, wherein a single generic record of master data is used to list all of the components and characteristics of all assets of a particular type. Existing systems are also oriented toward relationships with customers or other business partners, and not towards relationships with assets. Thus, in some cases, it is awkward for such systems to accurately reflect individual assets as they become more and more unique. SUMMARY In one aspect, the invention provides for individual data objects in an enterprise computing system, where the individual data objects represent unique assets of a family of assets. The individual data objects are used in addition to business partner master data, and also in some cases in addition to generic product master data. Thus, the enterprise computing system enables the modeling of business processes centrally around uniquely defined individual objects, while still retaining the advantages of a customer, or business partner, focused system. Such modeling is particularly useful for certain types of assets, such as assets that are highly configurable, have a high value, or have a long and complex lifecycle. In another aspect, the invention provides a computer system for processing individual data objects. The system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family, and includes, when first created, a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object based on user input, and that associates an attribute value with each added attribute. In various implementations, the computer system has one or more of the following features. The object family may identify the attributes that are permitted to be added to each individual object associated with the object family, and the executable software module may only add an attribute to an individual object when the attribute is identified by the object family as a permitted

attribute. Each individual object in the data repository may include an attribute value for each attribute of the individual object. The asset that an individual data object represents may be, for example, a tangible or intangible product, or a service. The computer system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family and includes, when first created, only a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and that associates an attribute value with each added attribute. In various implementations, the object family may identify the attributes that are permitted to be added to each individual object associated with the object family, and the executable software module only adds an attribute to an individual object when the attribute is identified by the object family as a permitted attribute. In another aspect, the invention provides a data repository including individual objects. Each individual object in the repository is associated with an object family and has a unique identifier attribute and attributes that are common with other individual objects that are associated with the same object family to which the object is associated. Each such attribute has a corresponding attribute value. In various implementations, each individual object in the repository pertains to a unique asset within an asset family. Additional attributes may be added to each individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and an attribute value may be associated with each added attribute. The repository may include asset information, information for a product, information for a service, and information for an intangible product. The repository may also include business partner information, with each individual object being capable of being associated with a business partner. In another aspect, the invention provides a data repository with product master data, business partner master data, and individual objects. The product master data includes product master records. Each product master record is associated with a particular type of product. The business partner master data includes business partner master records. Each business partner master record is associated with a particular business partner. As for the individual objects, each individual object a pertains to a unique asset within an asset family, b is associated with an object family corresponding to the asset family, c includes, when first created, a unique identifier and attributes, each attribute being associated with an attribute value, and d is associated with a particular product master record and a particular business partner master record. Additional attributes are added to an individual object in response to a temporal event having impact on the asset to which the individual object pertains, and attribute values are associated with each added attribute. In another aspect, the invention provides an integration system to integrate a computer system executing a software application with uniquely defined individual data objects. Each individual data object has firstly, attributes, secondly, allowable states for the attributes, and thirdly, allowable transitions between the allowable states. The integration system has an executable software module, wherein, in response to the integration system receiving, from the computer system executing the software application, information that requires an action to be performed on an identified one of the uniquely defined individual data objects, the executable software module produces a corresponding triggering event. The integration system also has data containers. The data containers are, firstly, for importing context data for the raised event, and secondly, for exporting context data returned by the identified individual data object in response to the generated triggering event. The integration system further has an executable event manager software module. This module, firstly, determines, based on the raised triggering event and any imported context data, an action to be performed on the identified individual data object, the event manager using a state manager to determine any transition in the allowable state of the identified individual data object that is required to be made in response to the generated triggering event. Secondly, the event manager software module executes the action, including any required state change, on the identified individual data object. In various implementations, the individual data objects may represent a configurable item whose configuration changes over the lifecycle of the item. The item may be a product such as an automobile, or a service, for example. The information received from the computer system may explicitly identify the triggering event, or it may do so implicitly. The individual object attributes may include an identifier that uniquely identifies an item

corresponding to the particular uniquely defined individual object. The unique identifier may be, for example, a vehicle identification number for an automobile. Possible triggering events include a data request, for example. The integration may further comprise a state manager adapter that interfaces the event manager with the state manager, which may be part of the integration system or alternatively may be in an external system called by the integration system. The state manager may include a state machine engine, and may include a repository of allowable states for individual data objects and a repository of allowable transitions between states. Upon execution of the action on the individual object, a data container may be populated with context data from the identified individual object. The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims. Like reference symbols in the various drawings indicate like elements. The group of systems includes an enterprise computing system , which is networked with a supplier network on a buy-side , and also with a customer network on a sell-side. In this example, the enterprise system includes a supply chain management SCM solution that links purchasing-related processes with the business procedures of suppliers. A customer relationship management CRM solution guarantees sales-related links with customers, and a supplier relationship management SRM solution guarantees supply-related links with suppliers.

ENTERPRISE COMPUTING WITH OBJECTS pdf

5: Enterprise Computing with Objects: From Client/Server Environments to the Internet | InformIT

Do you want to remove all your recent searches? All recent searches will be deleted.

We equip business leaders â€" across all major functions, in every industry and enterprise size â€" with the insights, advice and tools to achieve their mission-critical priorities and build the successful organizations of tomorrow. Gartner Top 10 Strategic Technology Trends for October 3, Trends Artificial intelligence, immersive experiences, digital twins, event-thinking and continuous adaptive security create a foundation for the next generation of digital business models and ecosystems. How do designers make cars safer? They treat them like a school of fish. Safe Swarm, recently unveiled by Honda , uses vehicle-to-vehicle communication to allow cars to pass information on to other cars in the vicinity. For example, alerts about an accident miles up the road could be relayed to cars miles back, enabling them to operate collaboratively and intelligently to avoid accidents and mitigate traffic. The evolution of intelligent things, such as collective thinking car swarms, is one of 10 strategic trends with broad industry impact and significant potential for disruption. How AI is seeping into virtually every technology and with a defined, well-scoped focus can allow more dynamic, flexible and potentially autonomous systems. Blending the virtual and real worlds to create an immersive digitally enhanced and connected environment. The connections between an expanding set of people, business, devices, content and services to deliver digital outcomes. AI Foundation The ability to use AI to enhance decision making, reinvent business models and ecosystems, and remake the customer experience will drive the payoff for digital initiatives through Lessons from Artificial Intelligence Pioneers Although using AI correctly will result in a big digital business payoff, the promise and pitfalls of general AI where systems magically perform any intellectual task that a human can do and dynamically learn much as humans do is speculative at best. Narrow AI, consisting of highly scoped machine-learning solutions that target a specific task such as understanding language or driving a vehicle in a controlled environment with algorithms chosen that are optimized for that task, is where the action is today. Watch now Trend No. Intelligent Apps and Analytics Over the next few years every app, application and service will incorporate AI at some level. AI will run unobtrusively in the background of many familiar application categories while giving rise to entirely new ones. AI has become the next major battleground in a wide range of software and service markets, including aspects of ERP. Intelligent apps also create a new intelligent intermediary layer between people and systems and have the potential to transform the nature of work and the structure of the workplace, as seen in virtual customer assistants and enterprise advisors and assistants. Augmented analytics is a particularly strategic growing area that uses machine learning for automating data preparation, insight discovery and insight sharing for a broad range of business users, operational workers and citizen data scientists. Intelligent Things Intelligent things use AI and machine learning to interact in a more intelligent way with people and surroundings. These things operate semiautonomously or autonomously in an unsupervised environment for a set amount of time to complete a particular task. Examples include a self-directing vacuum or autonomous farming vehicle. As the technology develops, AI and machine learning will increasingly appear in a variety of objects ranging from smart healthcare equipment to autonomous harvesting robots for farms. As intelligent things proliferate, expect a shift from stand-alone intelligent things to a swarm of collaborative intelligent things. In this model, multiple devices will work together, either independently or with human input. The leading edge of this area is being used by the military, which is studying the use of drone swarms to attack or defend military targets. Digital Twins A digital twin is a digital representation of a real-world entity or system. In the context of IoT, digital twins are linked to real-world objects and offer information on the state of the counterparts, respond to changes, improve operations and add value. With an estimated 21 billion connected sensors and endpoints by , digital twins will exist for billions of things in the near future. Potentially billions of dollars of savings in maintenance repair and operation MRO and optimized IoT asset performance are on the table, says Cearley. In the short term, digital twins offer help with asset management, but will eventually offer value in operational efficiency and insights into how products are used and how they can be improved. Cloud to the Edge Edge computing describes a computing topology in which information processing and

content collection and delivery are placed closer to the sources of this information. Connectivity and latency challenges, bandwidth constraints and greater functionality embedded at the edge favors distributed models. Enterprises should begin using edge design patterns in their infrastructure architectures â€" particularly for those with significant IoT elements. A good starting point could be using colocation and edge-specific networking capabilities. Cloud is a system where technology services are delivered using internet technologies, but it does not dictate centralized or decentralized service delivering services. When implemented together, cloud is used to create the service-oriented model and edge computing offers a delivery style that allows for executions of disconnected aspects of cloud service. Conversational Platforms Conversational platforms will drive a paradigm shift in which the burden of translating intent shifts from user to computer. The challenge that conversational platforms face is that users must communicate in a very structured way, and this is often a frustrating experience. A primary differentiator among conversational platforms will be the robustness of their conversational models and the API and event models used to access, invoke and orchestrate third-party services to deliver complex outcomes. Immersive Experience Augmented reality AR , virtual reality VR and mixed reality are changing the way that people perceive and interact with the digital world. Combined with conversational platforms, a fundamental shift in the user experience to an invisible and immersive experience will emerge. Application vendors, system software vendors and development platform vendors will all compete to deliver this model. Over the next five years the focus will be on mixed reality, which is emerging as the immersive experience of choice, where the user interacts with digital and real-world objects while maintaining a presence in the physical world. Blockchain Blockchain is a shared, distributed, decentralized and tokenized ledger that removes business friction by being independent of individual applications or participants. It allows untrusted parties to exchange commercial transactions. The technology holds the promise to change industries, and although the conversation often surrounds financial opportunities, blockchain has many potential applications in government, healthcare, content distribution, supply chain and more. However, many blockchain technologies are immature and unproven, and are largely unregulated. A practical approach to blockchain demands a clear understanding of the business opportunity, the capabilities and limitations of blockchain, a trust architecture and the necessary implementation skills. Identify the integration points with existing infrastructures, and monitor the platform evolution and maturation. Event-Driven Digital businesses rely on the ability to sense and be ready to exploit new digital business moments. Business events reflect the discovery of notable states or state changes, such as completion of a purchase order. Some business events or combinations of events constitute business moments â€" a detected situation that calls for some specific business action. The most consequential business moments are those that have implications for multiple parties, such as separate applications, lines of business or partners. With the advent of AI, the IoT, and other technologies, business events can be detected more quickly and analyzed in greater detail. Continuous Adaptive Risk and Trust Digital business creates a complex, evolving security environment. The use of increasingly sophisticated tools increases the threat potential. Continuous adaptive risk and trust assessment CARTA allows for real-time, risk and trust-based decision making with adaptive responses to security-enable digital business. Traditional security techniques using ownership and control rather than trust will not work in the digital world. This requires embracing people-centric security and empowering developers to take responsibility for security measures.

ENTERPRISE COMPUTING WITH OBJECTS pdf

6: Enterprise Distributed Object Computing - Wikipedia

Enterprise Computing with Objects: From Client/Server Environments to the Internet (OBT) by Yen-Ping Shan. Addison Wesley, This is an ex-library book and may have the usual library/used-book markings www.enganchecubano.com book has soft covers.

Overview[edit] Enterprise modelling is the process of building models of whole or part of an enterprise with process models , data models , resource models and or new ontologies etc. An enterprise includes a number of functions and operations such as purchasing, manufacturing, marketing, finance, engineering, and research and development. The enterprise of interest are those corporate functions and operations necessary to manufacture current and potential future variants of a product. For example, the use of networked computers to trigger and receive replacement orders along a material supply chain is an example of how information technology is used to coordinate manufacturing operations within an enterprise. For this purpose they include abstractions suitable for strategic planning , organisational re- design and software engineering. The views should complement each other and thereby foster a better understanding of complex systems by systematic abstractions. The views should be generic in the sense that they can be applied to any enterprise. Hence, enterprise models can be regarded as the conceptual infrastructure that support a high level of integration. One of the earliest pioneering works in modelling information systems was done by Young and Kent , [8] [9] who argued for "a precise and abstract way of specifying the informational and time characteristics of a data processing problem". They wanted to create "a notation that should enable the analyst to organize the problem around any piece of hardware ". Their work was a first effort to create an abstract specification and invariant basis for designing different alternative implementations using different hardware components. This led to the development of a specific IS information algebra. Ross , the one concentrate on the information view and the other on the function view of business entities. Specific methods for enterprise modelling in the context of Computer Integrated Manufacturing appeared in the early s. According to Fox and Gruninger from "a design perspective, an enterprise model should provide the language used to explicitly define an enterprise From an operations perspective, the enterprise model must be able to represent what is planned, what might happen, and what has happened. It must supply the information and knowledge necessary to support the operations of the enterprise, whether they be performed by hand or machine. Function modelling in systems engineering is a structured representation of the functions , activities or processes within the modelled system or subject area. The purpose of the function model are to describe the functions and processes, assist with discovery of information needs, help identify opportunities, and establish a basis for determining product and service costs. A functional perspectives is one or more perspectives possible in process modelling. Other perspectives possible are for example behavioural, organisational or informational. The main concept in this modelling perspective is the process, this could be a function, transformation, activity, action, task etc. A well-known example of a modelling language employing this perspective is data flow diagrams. The perspective uses four symbols to describe a process, these being: Illustrates transformation from input to output. Data-collection or some sort of material. Movement of data or material in the process. External to the modelled system, but interacts with it. Now, with these symbols, a process can be represented as a network of these symbols. This decomposed process is a DFD, data flow diagram. The data modelling process. Data modelling is the process of creating a data model by applying formal data model descriptions using data modelling techniques. Data modelling is a technique for defining business requirements for a database. It is sometimes called database modelling because a data model is eventually implemented in a database. A conceptual data model is developed based on the data requirements for the application that is being developed, perhaps in the context of an activity model. The data model will normally consist of entity types, attributes, relationships, integrity rules, and the definitions of those objects. This is then used as the start point for interface or database design. Business process modelling , not to be confused with the wider Business Process Management BPM discipline, is the activity of representing processes of an enterprise, so that the current "as is" process may be analyzed and improved in future "to be". Business process modelling is typically performed by business

analysts and managers who are seeking to improve process efficiency and quality. The process improvements identified by business process modelling may or may not require Information Technology involvement, although that is a common driver for the need to model a business process, by creating a process master. Change management programs are typically involved to put the improved business processes into practice. With advances in technology from large platform vendors, the vision of business process modelling models becoming fully executable and capable of simulations and round-trip engineering is coming closer to reality every day. Systems architecture[edit] The RM-ODP reference model identifies enterprise modelling as providing one of the five viewpoints of an open distributed system. Note that such a system need not be a modern-day IT system: Enterprise modelling techniques[edit] There are several techniques for modelling the enterprise such as Active Knowledge Modeling, [22].

7: Individual data objects in enterprise computing systems - SAP AG

The improvements of computer and network technology enable the control systems to handle massive information in the distributed computing environments like other.

What is claimed is: A computer system for processing individual data objects, the system comprising: The computer system of claim 1 wherein: The computer system of claim 1 wherein each individual object in the data repository includes an attribute value for each attribute of the individual object. The computer system of claim 1 wherein the asset is a product. The computer system of claim 1 wherein the asset is a service. The computer system of claim 1 wherein the asset is an intangible product. A computer system for processing individual data-objects the system comprising: The computer system of claim 7 wherein: The computer system of claim 7 wherein each individual object in the data repository includes an attribute value for each attribute of the individual object. A data repository including individual objects, wherein: The data repository of claim 10 wherein each individual object pertains to a unique asset within an asset family. The data repository of claim 11 wherein: The data repository of claim 11 wherein the repository includes asset information. The data repository of claim 13 wherein the asset information includes information for a product. The data repository of claim 13 wherein the asset information includes information for a service. The data repository of claim 13 wherein the asset information includes information for an intangible product. The data repository of claim 10 wherein: The data repository of claim 10 wherein each individual object is capable of including configuration information for the individual object. The data repository of claim 10 wherein each individual object is capable of being associated with history information for the individual object. The data repository of claim 10 wherein each individual object is capable of having location information for the geographical location of the individual object. The data repository of claim 10 wherein each individual object is capable of being associated with calendar information that includes dates and tasks, each task being associated with a particular date. A data repository comprising: An integration system that integrates a computer system executing a software application with uniquely defined individual data objects, wherein each uniquely defined individual data object has firstly, attributes, secondly, allowable states for the attributes, and thirdly, allowable transitions between the allowable states, the system comprising: The integration system of claim 23 wherein the individual data objects represent a configurable item whose configuration changes over the lifecycle of the item. The integration system of claim 24 wherein the item is a product. The integration system of claim 25 wherein the product is an automobile. The integration system of claim 24 wherein the item is a service. The integration system of claim 23 wherein the information received from the computer system explicitly identifies the triggering event. The integration system of claim 23 wherein the information received from the computer system implicitly identifies the triggering event. The integration system of claim 23 wherein the attributes include an identifier that uniquely identifies an item corresponding to a particular uniquely defined individual object. The integration system of claim 30 wherein the unique identifier is a vehicle identification number for an automobile. The integration system of claim 23 wherein possible triggering events include a data request. The integration system of claim 23 further comprising a state manager adapter that interfaces the event manager with the state manager. The integration system of claim 33 wherein the state manager is part of the integration system. The integration system of claim 23 wherein the state manager is a separate system called by the integration system. The integration system of claim 23 wherein the state manager includes a state machine engine. The integration system of claim 23 wherein the state manager includes a repository of allowable states for individual data objects and a repository of allowable transitions between states. The integration system of claim 23 wherein upon execution of the action on the individual object, a data container is populated with context data from the identified individual object. TECHNICAL FIELD [] This invention relates to computing master data that are structured as individual data objects wherein each such object uniquely defines a single asset, and also relates to integrating business computing processes with the individual data objects. BACKGROUND [] Customer relationship management CRM systems are designed to track relationships between a business and its customers which may be referred to as a type of

business partner , as well as the services or assets such as software, contracts, or property that form these relationships. Assets are typically tracked through records representing business partners that have relationships to certain assets. This is the case because CRM systems are designed to focus primarily on customers, or business partners. Additionally, records of these assets are sometimes kept in a tabular format, with a single unique identifier. Transactions and other events in CRM systems are typically initiated by determining the type of transaction desired. Parameters such as customer, cost and time are set, and then a suitable service or asset of the transaction must be found and selected, after which the appropriate modifications must be made to all of the various records that can represent the service or asset. In these areas, assets often become increasingly unique with the passage of time. Some CRM systems use product master data for representing assets, wherein a single generic record of master data is used to list all of the components and characteristics of all assets of a particular type. Existing systems are also oriented toward relationships with customers or other business partners, and not towards relationships with assets. Thus, in some cases, it is awkward for such systems to accurately reflect individual assets as they become more and more unique. SUMMARY [] In one aspect, the invention provides for individual data objects in an enterprise computing system, where the individual data objects represent unique assets of a family of assets. The individual data objects are used in addition to business partner master data, and also in some cases in addition to generic product master data. Thus, the enterprise computing system enables the modeling of business processes centrally around uniquely defined individual objects, while still retaining the advantages of a customer, or business partner, focused system. Such modeling is particularly useful for certain types of assets, such as assets that are highly configurable, have a high value, or have a long and complex lifecycle. The system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family, and includes, when first created, a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object based on user input, and that associates an attribute value with each added attribute. The object family may identify the attributes that are permitted to be added to each individual object associated with the object family, and the executable software module may only add an attribute to an individual object when the attribute is identified by the object family as a permitted attribute. Each individual object in the data repository may include an attribute value for each attribute of the individual object. The asset that an individual data object represents may be, for example, a tangible or intangible product, or a service. The computer system has a data repository for individual objects. Each individual object pertains to a unique asset within an asset family, is associated with an object family corresponding to the asset family and includes, when first created, only a unique identifier value, and capable of including additional attributes and attribute values. The computer system also has an executable software module that adds attributes to an individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and that associates an attribute value with each added attribute. Each individual object in the repository is associated with an object family and has a unique identifier attribute and attributes that are common with other individual objects that are associated with the same object family to which the object is associated. Each such attribute has a corresponding attribute value. Additional attributes may be added to each individual object in response to a temporal event having an impact on the unique asset to which the individual object pertains, and an attribute value may be associated with each added attribute. The repository may include asset information, information for a product, information for a service, and information for an intangible product. The repository may also include business partner information, with each individual object being capable of being associated with a business partner. The product master data includes product master records. Each product master record is associated with a particular type of product. The business partner master data includes business partner master records. Each business partner master record is associated with a particular business partner. As for the individual objects, each individual object a pertains to a unique asset within an asset family, b is associated with an object family corresponding to the asset family, c includes, when first created, a unique identifier and attributes, each attribute being associated with an attribute value, and d is associated with a particular product master record

and a particular business partner master record. Additional attributes are added to an individual object in response to a temporal event having impact on the asset to which the individual object pertains, and attribute values are associated with each added attribute. Each individual data object has firstly, attributes, secondly, allowable states for the attributes, and thirdly, allowable transitions between the allowable states. The integration system has an executable software module, wherein, in response to the integration system receiving, from the computer system executing the software application, information that requires an action to be performed on an identified one of the uniquely defined individual data objects, the executable software module produces a corresponding triggering event. The integration system also has data containers. The data containers are, firstly, for importing context data for the raised event, and secondly, for exporting context data returned by the identified individual data object in response to the generated triggering event. The integration system further has an executable event manager software module. This module, firstly, determines, based on the raised triggering event and any imported context data, an action to be performed on the identified individual data object, the event manager using a state manager to determine any transition in the allowable state of the identified individual data object that is required to be made in response to the generated triggering event. Secondly, the event manager software module executes the action, including any required state change, on the identified individual data object. The item may be a product such as an automobile, or a service, for example. The information received from the computer system may explicitly identify the triggering event, or it may do so implicitly. The individual object attributes may include an identifier that uniquely identifies an item corresponding to the particular uniquely defined individual object. The unique identifier may be, for example, a vehicle identification number for an automobile. Possible triggering events include a data request, for example. The integration may further comprise a state manager adapter that interfaces the event manager with the state manager, which may be part of the integration system or alternatively may be in an external system called by the integration system. The state manager may include a state machine engine, and may include a repository of allowable states for individual data objects and a repository of allowable transitions between states. Upon execution of the action on the individual object, a data container may be populated with context data from the identified individual object. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

8: Enterprise modelling - Wikipedia

Overview of Enterprise Computing: The Business View Agenda What is an enterprise? What constitutes an enterprise? Enterprise Computing with Objects, Addison.

9: Yen-Ping Shan (Author of Enterprise Computing with Objects)

Yen-Ping Shan is the author of Enterprise Computing with Objects (avg rating, 2 ratings, 0 reviews, published).

ENTERPRISE COMPUTING WITH OBJECTS pdf

Genitourinary Pathology Veterinary information on the treatment of parasites and diseases Organizational Collaboration, Culture, and Performance Political party evolution Lehninger principles of biochemistry indian edition What is a marine mammal? Transparency masters to accompany Systems analysis design methods The burden of knowledge and the death of the renaissance man Teaching Low Achieving And Disadvantaged Students Auckland (Panarama 360) Combat Aircraft of WWII 1938-3 Celts and Stonehenge The constitution of australia a contextual analysis Advances in fuel cells Events That Changed the World 1980-2000 Safety for your children Research writing simplified 4th edition Introduction : reading Herodotus, reading book 5 Elizabeth Irwin and Emily Greenwood Distance formula word problems worksheet My Life as a Smashed Burrito With Extra Hot Sauce (The Incredible Worlds of Wally McDoogle #1) The Man with the Clubfoot Homotopic Paths and the Fundamental Group Photomicrography of paper fibers THE COMPLETE CAT ORGANIZER Tales of the Northwoods Traditional Jewish faith and practice Big Book of Papercrafts Hurricane Katrina devastates New Orleans. Simple living in history Shepards Ohio legal filing directory The complete sportsmans encyclopedia Assessment of teamwork skills via a teamwork questionnaire Harold F. ONeil, Jr. . [et al.] Collectors Guide to Raphael Tuck Sons Universities and Globalization The Sailors Sketchbook Build Your Own Army of Web Bots Within 24 Hours (Army of Web Bots Series, 1) Cottage piety exemplified Conscience and Other Virtues Security in fixed and wireless networks The story of Fuzzypeg the Hedgehog