SUSTAINING A FLEET OF CONFIGURATION-CONTROLLED ASSETS

CROSS REFERENCE TO RELATED APPLICATIONS

The present application claims priority to co-pending U.S. provisional patent application entitled "SUSTAINING A FLEET OF CONFIGURATION-CONTROLLED ASSETS" having serial number 60/740,351, filed November 29, 2005.

BACKGROUND OF THE INVENTION

The present invention generally relates to integrating a plurality of applications for sustaining the readiness of a fleet of assets, such as vehicles and/or aircraft and, more particularly, relates to a method of integrating a plurality of applications by creating different system layers to facilitate replacement of each application and prevent modification of the rest of the applications.

The fleet vehicle industry and customers have identified integrated information and decision support as key to both fleet support and vehicle readiness, with war-fighter readiness as a specific example. Both customers and industry are investing in system capabilities to leverage automation and decision support in the sustainment environment. Because fleet operations can be critical and each customer has unique needs, there is a need for robust, flexible, and tailored systems and processes.

Generally, in a fleet management system, applications perform particular tasks such as integrated vehicle health management, maintenance management, materials management, engineering analysis, and training management. Each application may include both processes and support resources such as manuals, historical data, and personnel data. In operation, a fleet management system exchanges information with different applications, as needed, in order to analyze the current condition of the fleet, make decisions, and create course of action.

Typically, applications designed for the sustainment of the fleet include existing, or "legacy," applications and new applications with capabilities necessary to adapt to the changing operational needs of the fleet. The entities which comprise the fleet may be independent of one another but still use the same suite of applications. In other words, integrated fleet management

systems often need to accommodate multiple and distinct fleets of assets. Historically, however, these systems are not flexible enough to accommodate easy modification of the underlying suite of applications.

For example, some conventional systems for the sustainment of a fleet of configuration-controlled assets consist of an integration of domain-specific sustainment applications built on a point-to-point architecture. This architectural approach, as opposed to a more flexible design based on an open definition of system-to-system interfaces, requires redevelopment of each system interface when new integration needs arise, thus resulting in increased development costs. In addition, conventional systems do not offer domain-specific knowledge optimization.

Accordingly there is a need in the industry to address how to develop a fleet management system to simplify future modifications of the system and reduce time and cost associated with each modification. There is also a need to address the aforementioned and other deficiencies and inadequacies.

BRIEF DESCRIPTION OF THE INVENTION

In one aspect, a computer-implemented method for sustaining a fleet of configuration-controlled assets is provided. The method includes integrating a plurality of previously separate systems with reusable business transactions associated with managing a fleet of assets wherein functional capabilities of the previously separate systems are defined by the reusable business transactions. The method also includes receiving operational data associated with at least one asset of the fleet, integrating said data per customer requirements, packaging and distributing said data to the plurality of systems, and installing and distributing said data to one or more applications of the plurality of systems utilizing at least one of the reusable business transactions.

In another aspect, a computer-implemented system for integrating a plurality of applications is provided. The system comprises an integration manager operative to manage the insertion and removal of applications to and from the plurality of applications, an integration gateway operative to interface reusable business transactions and the plurality of applications, one or more application adapters interfacing the plurality of applications with the integration gateway, an intelligent transaction manager extract operative to extract and install reusable business transactions, an anomaly tracking and reporting system, and a performance metrics

portal for aggregating system performance data, wherein the plurality of applications are integrated via the reusable business transactions thereby allowing for insertion and removal of previously separate applications without redevelopment.

In still another aspect, a computer program product comprising a computer-readable medium having control logic stored therein to enable a computer to sustain a fleet of configuration-controlled assets is provided. The control logic comprises computer-readable program code for causing the computer to integrate a plurality of previously separate systems with reusable business transactions associated with managing a fleet of assets, to receive operational data associated with at least one vehicle of the fleet, to integrate the operational data per customer requirements, to package and distribute the operational data to the plurality of systems, and to install and distribute the operational data to one or more applications of the plurality of systems utilizing at least one of the reusable business transactions.

In yet another aspect, a system for sustaining a fleet of vehicles is provided. The system comprises a plurality of applications associated with the management of a fleet of mobile platforms, a plurality of logical subsystems, each of which is associated with a respective one of the plurality of applications, and a plurality of gateways, each of which connects one of the plurality of applications to a respective one of the plurality of logical subsystems and converts the data flow between the applications and the subsystems.

Additionally, a method for managing the insertion and removal of previously separate domain applications that have been integrated with one another is provided. The method comprises configuring an extended application interface to connect the domain applications together, providing a gateway operative to interface reusable transactions with the domain applications, operating a transaction manager to extract and install the reusable transactions, and implementing an application adapter for each new domain application to be added, with the application adapter configured to interface the new domain application to the extended application interface.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram illustrating aspects of a networked operating environment.

FIG. 2 illustrates computing system architecture for a sustainment data system (SDS) server computer.

FIGS. 3a-3b illustrate a block diagram of an SDS integration platform based on the computer of FIG. 2.

FIG. 4 illustrates an operational flow performed in sustaining configuration-controlled vehicles.

FIG. 4a illustrates a simplified version of FIG. 4.

FIG. 5 illustrates aspects of a networked operating fleet management system.

FIGS. 5a-5c illustrate a block diagram of a service-oriented architecture (SOA) implemented by an SDS integration platform.

DETAILED DESCRIPTION OF THE INVENTION

As described briefly above, embodiments of the present invention provide an architectural approach to an integrated fleet management system where the definition of the functional capability required by the overall system is kept independent from the application used to support that capability. This approach facilitates future modifications to the system by allowing individual applications to be replaced without affecting the overall capability of the system. In the following detailed description, references are made to accompanying drawings that form a part hereof, and in which are shown by way of illustration specific embodiments or examples. These illustrative embodiments may be combined, other embodiments may be utilized, and structural changes may be made without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims and their equivalents.

Referring now to the drawings, in which like numerals represent like elements through the several figures, aspects of the present invention and the illustrative operating environment will be described. FIGURES 1-5c and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the embodiments of the invention may be implemented. While the invention will be described in the general context of

program modules that execute in conjunction with a BIOS program that executes on a personal or server computer, those skilled in the art will recognize that the invention may also be implemented in combination with other program modules. It should be noted that hereinafter, the words module and application will be used interchangeably.

Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Aspects of the invention may be implemented as a computer process, a computing system, or as an article of manufacture such as a computer program product or computer-readable medium. The computer program product may be a computer storage media readable by a computer system and encoding a computer program of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process.

These and various other features as well as advantages, which characterize the present invention, will be apparent from a reading of the following detailed description and a review of the associated drawings.

Embodiments of the present invention disclose a sustainment data system (SDS). The SDS is a net-centric system-of-systems support architecture that includes support system elements to provide decision-aided, seamless integration and management of support resources and processes. This net-centric system functionality is based upon access to a network. Use of the SDS optimizes cost, availability, and capability of supported systems through integration with both customers and partners of the entity using the SDS. By developing and implementing

common processes to integrate disparate systems in a reusable fashion, the SDS reduces systems development time and allows its integration efforts to be extensible to the commercial world.

Referring now to FIG. 1, a schematic diagram illustrating aspects of a networked operating environment 100 utilized in an illustrative embodiment of the invention will be described. As shown in FIG. 1, the networked environment 100 includes an SDS server 102 in communication with a backend server 104. The backend server 104 includes an intelligent transaction manager (ITM) 103 that contains reusable business transactions defined via a collaborative development process including input at one or more workstations 105. Capabilities of the previously separate systems are defined by reusable business transactions. The SDS server 102 implements reusable business transactions from the ITM 103 residing on the backend server 104. Data associated with a fleet of configuration-controlled vehicles 112a-112n may be transferred to the SDS server 102 via a portable disk or a vehicle communications bus 113.

The SDS server 102 is accessible to personal computer (PC) 108 and/or a laptop 107 via a SDS network 117. Examples of such a network include the Internet or an intranet. The networked environment 100 also includes an external data system 114, such as an engineering data system and/or a logistics data system for updating and receiving data associated with engineering and/or logistics analysis. Additional details regarding the SDS server 102 will be described below with respect to FIGs. 2-3.

FIG. 2 illustrates a computing system architecture for the SDS server 102 utilized in an illustrative embodiment of the invention. The SDS server 102 includes a central processing unit (CPU) 210, a system memory 202, and a system bus 252 that couples the system memory 202 to the CPU 210. The system memory 202 includes read-only memory (ROM) 206 and random access memory (RAM) 204. System memory may also include non-volatile memory (not shown) that is not ROM. A basic input/output system 208 (BIOS), containing the basic routines that help to transfer information between elements within the SDS server 102, such as during start-up, is stored in ROM 206. The SDS server 102 further includes a mass storage device (MSD) 214 for storing an operating system 216 such as WINDOWS XP, from MICROSOFT CORPORATION of Redmond, Washington, an application server 217, such as those compliant with J2EE, (Java 2, Enterprise Edition), from SUN MICROSYSTEMS INC. of Santa Clara, California, and an SDS integration platform 221 for integrating previously separate systems or applications associated with the sustainment of assets, such as the fleet of vehicles 112a-112n.

The SDS integration platform 221 may occupy one or more nodes or servers. The SDS platform 221 includes an application integration section 224, an information integration section 227, an information utility integration section 228, and a communication integration section 230. Additional details regarding the SDS integration platform 221 will be described below with respect to FIGS. 3-4.

It should be appreciated that the MSD 214 may be a redundant array of inexpensive discs (RAID) system for storing data. The MSD 214 is connected to the CPU 210 through a mass storage controller (not shown) connected to the system bus 252. The MSD 214 and its associated computer-readable media, provide non-volatile storage for the SDS server 102. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or RAID array, it should be appreciated by those skilled in the art that computer-readable media can be any available media that can be accessed by the CPU 210.

The CPU 210 may employ various operations, discussed in more detail below with reference to FIG. 4 to provide and utilize the signals propagated between the SDS server 102 and networked data systems (FIG. 1). The CPU 210 may store data to and access data from MSD 214. Data is transferred to and received from the MSD 214 through the system bus 252. The CPU 210 may be a general-purpose computer processor. Furthermore as mentioned below, the CPU 210, in addition to being a general-purpose programmable processor, may be firmware, hard-wired logic, analog circuitry, other special purpose circuitry, or any combination thereof.

According to various embodiments of the invention, the SDS server 102 operates in a networked environment, as shown in FIG. 1, using logical connections to remote computing devices via network communication, such as an Intranet, or a local area network (LAN). The SDS server 102 may connect to the network 117 via a network interface unit 212. It should be appreciated that the network interface unit 212 may also be utilized to connect to other types of networks and remote computer systems.

A computing system, such as the SDS server 102, typically includes at least some form of computer-readable media. Computer readable media can be any available media that can be accessed by the SDS server 102. By way of example, and not limitation, computer-readable media might comprise computer storage media and communication media.

Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, disk drives, a collection of disk drives, flash memory, other memory technology or any other medium that can be used to store the desired information and that can be accessed by the SDS server 102.

Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above should also be included within the scope of computer-readable media. Computer-readable media may also be referred to as a computer program product.

FIGS. 3A-3B illustrate a block diagram of the SDS integration platform 221 illustrated in FIG. 2 according to an illustrative embodiment of the invention. The SDS integration platform 221 accomplishes its goal of application integration via the technical implementation of business-level, application-to-application message contracts. The main benefit of this approach is that substituting new applications requires only an implementation of an application adapter 308 for each new application instead of complete redevelopment effort to re-implement a point-to-point connection.

The SDS integration platform 221 is a net-centric system-of-systems support architecture that includes support system elements to provide decision-aided, seamless integration, and management of support resources and processes to optimize cost, availability, and capability of supported systems. The SDS approach is to provide a lightweight flexible integration environment to connect existing systems with intelligent programmable business logic contained in modular application gateway connectors, such as the application adapters 308, thru verified transactions with anti-spoofing capabilities. These verified transactions may use industry standard message formats (Open Application Group Integration Specification Business Object Document (OAGIS BOD)) to provide additional value through the analysis of Integrated Vehicle

Health Management (IVHM) data, increased efficiencies for supply chain management (SCM), optimization of maintenance planning and scheduling, or other sustainment-specific analysis activities.

This approach enables, with minimal effort, the substitution of different off-the-shelf software packages into the solution universe while not interfering with the underlying system functionality. The SDS application approach combines existing message-oriented middleware (MOM) and Service-Oriented Architecture (SOA) with modular application gateway connectors, and secure, industry-standard formatted messages. This provides a secure, fully integrated, and extensible application environment that may be applied in any arena requiring configuration-managed maintenance and automated advanced planning for optimized results. This application framework may be extended outside of this arena. The purpose of SDS is to build a sustainment solution, using tools inside of the constraints that a customer may levy on the sustainment solution provider. These constraints may include cost, existing infrastructure, and security.

Information Integration

The SDS integration platform 221, through its system-of-systems approach, provides an extended application bus allowing applications to be connected together through the use of a lightweight MOM, such as an application integration manager 309 operative to manage the insertion and removal of applications to and from the integrated applications. The information integration section 227 also includes an integration gateway 302 operative to interface reusable business transactions with applications, one or more application adapters 308 interfacing a plurality of applications with the integration gateway 302, and an intelligent transaction manager extract 304 operative to extract and utilize reusable business transactions from the intelligent transaction manager 103.

Additionally, the information integration section 227 includes an anomaly tracking and reporting system 310 and a performance metrics portal 307 for aggregating system performance data. Both the anomaly tracking and reporting system 310 and the performance metrics portal 307 are accessible to any application integrated into the SDS integration platform 221. Each application is integrated via a reusable business transaction, thus allowing for insertion and removal of previously separate applications without the need to redevelop the interface for that application. The reusable business transactions can surface previously latent features in one or

more of the integrated applications. Also, while reusable business transactions are documented in eXtensible Markup Language (XML), storage and representation of the reusable business transactions are enabled to extend to alternative technologies.

Information Utility Integration

The information utility integration section 228 includes a single sign-on authentication module 330 operative to manage users of the system via a central authoritative logon point for all the integrated applications. The single sign-on module 330 is also operative to identify and authenticate a user, and issue role-based credentials for the user to use during an engagement, such as from a time of logon through a time of logout with a maximum time period. Additionally, the information utility integration section 228 includes a data consolidation and distribution service 332, an audit service 334, a message validation repository service 335, and a system integration and business logic service 337. Still further, the information utility integration section 228 includes a security utilities, authorization, and encryption service 338 and an off node connector 340. Each utility service is connected to an application adaptor 308. The security utility service 338 includes authorization and encryption utilities. The authorization utility service verifies reusable business transactions against a primary real-time repository and/or a runtime extract of the repository. The encryption utility service is used to encrypt the payload of the messages. The encryption utility service is replaceable to meet the varied levels need for security and export control.

The system integration business logic service 337 is operative to instantiate a service-oriented architecture (SOA), permit interconnection of previously separate applications through the reusable business transactions, and prohibit direct use of external interfaces controlled by the system. This is accomplished by executing program code through this service; to enable the exchange of information, the business logic service may execute a programming language similar to Business Process Execution Language (BPEL). Additionally, the SDS integration platform 221 is operative to implement modular advanced services for the integrated applications, thus adding capability to the system via a utility application that leverages the reusable business transactions when a customer needs capability beyond existing capability of the application. Still further, the SDS integration platform 221 is also operative to implement an integrated transmission test capability to capture transmitted reusable business transactions in an audit log.

Application Integration

An application integration section 224 may include a variety of off-the-shelf and/or proprietary applications such as a mission planning system 312, a supply chain management system 314, a training management system 315, and a vehicle configuration and state database 317. Additionally, the application integration section 224 includes an optimized resource planning service 318, a technical publications system 320, a maintenance management system 322, and an integrated vehicle health management system 324. The vehicle health management system 324 may communicate with one or more vehicles via the vehicle communications bus 113 connecting up or receiving media from a vehicle that is plugged into the computer. Data may be retrieved via a direct cable connection with the vehicle, or may be retrieved via a removable computer storage medium such as a floppy disk, PCMCIA card, or other portable device.

Communication Integration

The communication integration section 230 includes protocol services 352 including TCP/IP and a net-centric communications services and discovery module 350 including a server application, such as a Java compliant application server.

SDS Business End to End Sustainment Integration

Referring now to FIGS. 1-4a, operational flow 400 performed in sustaining configuration-controlled vehicles or assets will be described. FIG. 4 is an illustrative routine or operational flow performed in sustaining configuration-controlled assets according to an illustrative embodiment of the invention and FIG 4a is a simplified version of FIG 4. When reading the discussion of the routines presented herein, it should be appreciated that the logical operations of various embodiments of the present invention are implemented (1) as a sequence of computer-implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations illustrated in FIGS. 4 and 4a, and making up the embodiments of the present invention described herein are referred to variously as operations, structural devices, acts, or modules. It will be recognized by

one skilled in the art that these operations, structural devices, acts, and modules may be implemented in software, in firmware, in special-purpose digital logic, and any combination thereof without deviating from the spirit and scope of the present invention as recited within the claims set forth herein.

The SDS server 102 is designed to integrate applications in the logistics sustainment support (LSS) domain. The SDS server 102 addresses the integration of maintenance systems with supply chain, with vehicle operational data, with qualification of personnel, and with parts and support equipment systems. The SDS server 102 does not limit the systems being integrated to a single vendor or source. The SDS integration approach documents the integration contracts for transactions between the LSS business systems. For instance the SDS server 102 utilizes a method of integrating previously separate domain applications. This represents a method to recognize, document, optimize, and codify a business process in areas of interest between previously separate applications. The SDS server 102 decouples system capability from functional assignment within each business system via the designed business contract, negotiated business transaction between systems, and system assignment to a contract. This allows for the substitution of applications should an application become unavailable.

The operational flow 400 begins with the SDS server 102 integrating technical data received from an engineering data system according to customer requirements. At operation 404, the SDS server 102 packages and distributes the integrated technical data thru a distribution system of the SDS integration platform 221. Then, at operation 407, the SDS server 102 installs and distributes the technical data to one or more applications of the integrated systems utilizing one or more reusable business transactions. The SDS server may also render a display of the operational data at operation 408.

As an article, aircraft, or vehicle completes a mission or an operation where usage occurs, the SDS server 102 receives sensor data and a manual debrief at operation 410 via the integrated vehicle health management system (IVHM) 324. Also at operation 410, the SDS server 102 transfers fault or exceedence data to the maintenance management system 322. Next at operation 412, the SDS server 102 evaluates and determines maintenance needs of the vehicle for which data was received to create a work order. Then at operation 414, the SDS server 102 determines whether parts are available within a supply chain to fulfill the work order at the supply chain management (SCM) system 314.

Meanwhile at operation 417, the SDS server 102 determines an optimal schedule to perform the work order via the optimized resource planning service 318. The SDS server 102 may also, at operation 420, asynchronously determine whether a trained resource is available to perform the work order via the training management system. The SDS server 102 may also plan a mission for one of more vehicles via the mission planning system 312 and optimize the plan via the optimized resource planning service 318.

Once performance of the workorder is completed, the SDS server 102 records the work performed at operation 424 via the performance metrics portal 307, aggregates metrics data associated with sustaining the fleet, and calculates one or more key performance indicators (KPIs) based on the metrics data collected. The KPIs have been previously identified by a customer. The SDS server 102 may also asynchronously track and report on anomalies detected via one or more of the integrated systems at operation 422.

Next at operation 427, the SDS server 102 forwards metrics data or the KPIs for engineering system analysis and feedback. The data may be forwarded to an external data system such as the data system 114. Here the metrics data or KPIs are analyzed for reliability, maintainability, and performance trends related to the fleet and its assets. The SDS server 102 may also forward metrics data or the KPIs for logistical system analysis and feedback at other operations. Control passes back to engineering operations at connector 1 where the business process resumes.

It should be appreciated that the SDS server 102 manages users of the integrated applications or systems via a central authoritative logon for all of the systems. The SDS server 102 receives logon inputs from one or more users, identifies and authenticates the users, and issues a role-based credential for each user to be used during a user session accessing one or more of the integrated systems. A user's role is assigned by this mechanism and persisted across all systems. All of the integrated systems tie into a single management dashboard presentation that displays KPIs and provides decision support for the various levels of a customer's organization.

Referring now to FIG. 5, there is shown a schematic diagram illustrating aspects of a networked fleet management system 500 of this invention. The fleet management system 500 utilizes a sustainment data system (SDS) 502. The SDS is a network-centric system-of-systems

support architecture that provides seamless integration between different system applications 504 in order to automate the fleet management process. It should be noted that hereinafter the sustainment data system (SDS) and the SDS network are used interchangeably. The SDS network 502 connects various applications 504 such as integrated vehicle health management 504a, maintenance management 504b, materials management 504c, training management 504d, and technical data management 504e.

FIGs 5a-5c illustrate a service oriented architecture SOA 505 implementation of the SDS network 502 of FIG 2. The SOA 505, in FIG. 5a, includes a business layer 506, a gateway layer 507, and an application layer 508. The business layer 506 includes logical subsystems or components of each integrated system-of-systems, such as logical subsystems A1 509 and B1 510. Physical communication between the integrated systems occurs via the business layer 506 utilizing reusable business transactions such as a message 511. The reusable business transactions are represented by a set of messages 508 which constitute a common format for communication between the subsystems. For a more detailed explanation of the reusable transactions, reference is made to U.S. Patent Application Serial Number 11/343,137, filed January 30, 2006, see Appendix A.

The gateway layer 507 includes gateways, such as gateways A2 512 and B2 514, which connect logical subsystems of the business layer 506 such as subsystems A1 509 and B1 510 to the applications of the application layer 508 such as applications A3 515 and B3 517 respectively. For instance, the logical subsystem B1 510 is connected to the application B3 517 via the gateway 514. It should be noted that the business layer 506 uses a common format defined by the messages 508. However, the application layer 508 may require a plurality of formats each of which associated with one of its applications. In other words, multiple applications may use the same gateway and support the same logical subsystem. The supporting applications are transparent to the end user of the system, as illustrated in Figure 5C, described herein.

The gateway 514 converts the data flow between the logical subsystem B1 510 and the application B3 517 to allow the application layer 508 and the business layer 506, which require different formats, to communicate with each other. For example, the gateway 514 converts messages from logical subsystem B1 510, such as the message 508, to a format used by the

application B3 517 and visa versa. Hereinafter, convert shall mean performing one or more of the following actions: translating, reformatting, repackaging, and filtering.

In the system of FIG. 5a, communication from the application A3 515 to the application B3 517 is routed via the business layer 506 through the gateways A2 and B2 to facilitate flexible changes to the application layer 508 with minimal changes to the business layer 506. Additional details regarding facilitating changes to the application layer 508 will be described below with respect to FIG. 5b.

FIG. 5b illustrates the SOA 505' when a change to the application B3 517 occurs. When the application B3 517 is replaced by a new application B3 522, a new gateway B2 520 is also generated to convert the communication format to and from the new application B3 522. For instance, the new gateway B2 converts the message from the logical subsystem B1 517, such as the message 508, to a format used by the new application B3. Similarly, when the new application B3 produces a result, the new gateway B2 converts the result to be presented in the form of a defined message such as message 508. Applications can be replaced, without affecting the overall SDS system of systems, by selecting a new application, such as the new B3, implementing a new gateway, such as the new B2, and attaching the new gateway to the subsystem B1.

FIG. 5c illustrates the SOA 505" according to another embodiment of the present invention. A logical subsystem, such as the logical subsystem B1 510', may use a single application within the SOA 505". However, the logical subsystem B1 510' may also use a combination of applications, such as the applications 517a-517c, where each application is mediated by a gateway, such as gateways 514a-514c.

Thus, the present invention is presently embodied as methods, systems, computer program products or computer readable mediums encoding computer programs for sustaining a fleet of configuration-controlled vehicles.

The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Therefore, while the invention has been described in terms of various specific embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the claims.

REPRESENTING BUSINESS TRANSACTIONS

Technical Field

[001] The present invention generally relates to representing business transactions and, more particularly, relates to methods, configurations, computer-readable mediums, and systems for verifying business object documents via a relational repository.

Background

[002] A business object document (BOD) can be spoofed (a counterfeit BOD sent without a mechanism to validate its form) without validation. Previously, hard coded rules that perform validation become a maintenance and accuracy issue. This problem also becomes one of scale as the number of BODs used in an enterprise or industry grows into the hundreds and more with the application of BOD technology to electronic data interchange (EDI) like transactions.

[003] Some conventional systems use a repository that stores XML as a single structure including an OAGIS BOD list at a gross level. The repository is XML oriented but it does not handle the storage and management of BODs. These conventional systems support the transposition from older to new versions of intelligent transport messages (ITMs), but not at runtime or for ITM translations between versions. These conventional systems also do not support an integrated editing and logical transposition of business objects.

[004] For instance, Text Editors such as XMLSPY, POSEIDON, and MICROSOFT OFFICE WORD are all types of Text Editing tools. These text editors do not support a storage methodology. They support the transposition from older to new versions of the ITMs, but not at runtime. Similarly, Database Storage such as ORACLE, MICROSOFT ACCESS, and FOXPRO are all types of database storage tools. Although relational database technology may be supported by the above tools in order to store and manage the ITMs, these tools do not support a integrated editing and logical transposition of business objects.

[005] Additionally, XML-based Graphical Editors such as XMLSPY, POSEIDON, MICROSOFT OFFICE WORD, and MICROSOFT VISIO are all types of XML-based Graphical Editing tools. These tools do not support a storage methodology. They support the transposition from older to new versions of the ITMs, but not at runtime. Similarly, XMLValidators such as XMLSPY, POSEIDON, MICROSOFT OFFICE WORD, and MICROSOFT VISIO are all types of XML Validation tools. These tools do not support a storage methodology. They support the validation or transposition from older to new versions of the ITMs, but not at runtime. Also, Versioning Tools such as CVS, MICROSOFT SOURCESAFE, and IBM CLEARCASE are all types of versioning tools. These tools do not supply a runtime component that allows for ITM

translations between versions. They also do not allow for the transposition from older to new versions of the ITMs.

[006] Accordingly there is an unaddressed need in the industry to address the aforementioned and other deficiencies and inadequacies.

Brief Description of the Drawings

[007] FIG. 1 is a schematic diagram illustrating aspects of a networked operating environment and business object document verification by relational repository (BODVRR) architecture utilized in an illustrative embodiment of the invention;

[008] FIG. 2 illustrates a data content architecture for an ITM utilized in an illustrative embodiment of the invention;

[009] FIG. 3 illustrates a schematic diagram of a BODVRR and gateway platform according to an illustrative embodiment of the invention;

[010] FIG. 4 illustrates an operational flow performed in representing business transactions according to an illustrative embodiment of the invention;

[011] FIG. 5 illustrates a display utilized for viewing inputs and output of a repository according to an illustrative embodiment of the invention; and

[012] FIGS. 5a-5c illustrate a block diagram of a service-oriented architecture (SOA) implemented by an SDS integration platform according to illustrative embodiments of the invention.

Detailed Description

[013] As described briefly above, embodiments of the present invention provide methods, systems, configurations, and computer-readable mediums for representing business transactions. In the following detailed description, references are made to accompanying drawings that form a part hereof, and in which are shown by way of illustration specific embodiments or examples. These illustrative embodiments may be combined, other embodiments may be utilized, and structural changes may be made without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims and their equivalents.

[014] Referring now to the drawings, in which like numerals represent like elements through the several figures, aspects of the present invention and the illustrative operating environment will be described. FIGURES 1-3 and the following discussion are intended to

provide a brief, general description of a suitable computing environment in which the embodiments of the invention may be implemented. While the invention will be described in the general context of program modules that execute in conjunction with a BIOS program that executes on a personal or server computer, those skilled in the art will recognize that the invention may also be implemented in combination with other program modules.

[015] Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[016] Aspects of the invention may be implemented as a computer process, a computing system, or as an article of manufacture such as a computer program product or computer-readable medium. The computer program product may be a computer storage media readable by a computer system and encoding a computer program of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process. These and various other features as well as advantages, which characterize the present invention, will be apparent from a reading of the following detailed description and a review of the associated drawings.

[017] An embodiment of the present invention is a computer program that organizes a collection of Business Object Documents (BODs) by the taxonomy of the verbs and nouns that make up the BOD. This program provides for the approval cycle between those requesting a standard BOD, the search on the complete BOD name, and the verb or the noun. In addition to the description of the BOD, the construct of the BOD is able to be stored in a variety of formats including xml, uml, and other binary formatted data in a blob format. This program also allows for the verification of a BOD in an application against the certified format in the BOD repository.

[018] Referring now to a schematic diagram illustrating aspects of a networked operating environment 100 and business object document verification by relational repository (BODVRR) architecture 102 utilized in an illustrative embodiment of the invention, will be described. As shown in FIGURE 1, the networked environment 100 includes sustainment data systems (SDS)

112, a server 110, an SDS intelligent transport message (ITM), a workstation 104, and a printer 114. A method for representing business transactions, creating & managing those representations, and for exchanging and processing of those transactions on one or more computers linked by a communications connection is illustrated. This communications connection may include both point to point and network connectivity of both wireless and wired technology.

[019] The operating environment 100 includes one or more computers, a computer network, a keyboard 101, and a display 103. Internally, the BODVRR 102 is divided into the following components: human-machine interface (HMI) 115, create, edit, delete capability (CED) 118, search, sort, report capability (SSR) 122, and graphic rendering 127. The BODVRR 102 also includes assemble, process, form validation capability (V&V) 105, packaging and export capability 120, versioning and translation capability 124, and runtime component for performing validation 130. The BODVRR 102 is an integrated development, storage, and control environment for intelligent transport contracts (ITCs) and ITMs. Additional details regarding the BODVRR 102 will be described below with respect to FIG. 3.

[020] The application of relational database technology and business process methodology OAG suggested BOD approval process. BODVRR stores information in an external relational database for management purposes. This database 102 includes an automated approval process, search on name, verb or noun capability and the associated storage of the technical content of the BOD. The search, retrieval and management of stored version controlled BODs in the relational database allows them to be verified without going thru the user interface with appropriate security permission to the database. The automated creation of an XML schema for validation does not require the BODVRR 102 to be present in each system sending BODs. Additionally, the search capability 122 is available to locate content within the BOD Data fields or Binary Data.

[021] It should be appreciated that the BODVRR 102 may be a redundant array of inexpensive discs (RAID) system for storing data. The BODVRR 102 is connected to a CPU through a mass storage controller (not shown) or network. The BODVRR 102 and its associated computer-readable media, provide non-volatile storage. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or RAID array, it should be appreciated by those skilled in the art that computer-readable media can be any available media that can be accessed by the CPU.

[022] The CPU may employ various operations, discussed in more detail below with reference to FIG. 5 to provide and utilize the signals propagated between the BODVRR 102 and the SDS systems 112 (FIG. 1). The CPU may store data to and access data from BODVRR 102.

The CPU may be a general-purpose computer processor. Furthermore as mentioned below, the CPU, in addition to being a general-purpose programmable processor, may be firmware, hard-wired logic, analog circuitry, other special purpose circuitry, or any combination thereof.

[023] According to various embodiments of the invention, the BODVRR 102 operates in a networked or point to point environment, as shown in FIG. 1, using logical connections to remote computing devices via point to point or network communication, such as an Intranet, or a local area network (LAN). The BODVRR 102 may connect to the network via a network interface unit. It should be appreciated that the network interface unit may also be utilized to connect to other types of networks and remote computer systems.

[024] A computing system, such as the BODVRR 102, typically includes at least some form of computer-readable media. Computer readable media can be any available media that can be accessed by the BODVRR 102. By way of example, and not limitation, computer-readable media might comprise computer storage media and communication media.

[025] Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, disk drives, a collection of disk drives, flash memory, other memory technology or any other medium that can be used to store the desired information and that can be accessed by the central server 104.

[026] Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above should also be included within the scope of computer-readable media. Computer-readable media may also be referred to as computer program product.

[027] FIG. 2 illustrates a data content architecture for an ITM 107 utilized in an illustrative embodiment of the invention. An ITC is a business transaction contract to exchange data between two applications in a service-oriented architecture, importing data from outside of the system of systems (SOS) SOA and exporting from the SOS SOA. This contract will establish the following characteristics:

- definition of data being exchanged;

- translations required (language);
- any conversions required (to adapt Message Version);
- intermediate business processing of mutual benefit; and
- security protocols required (encryption, user level authentication).

[028] Referring still to FIG. 2, the ITM 107 includes data content based on an ITC between two or more systems, and will be represented by business transaction data formatted as XML-structured data 210, business transaction logic 204, business transaction constraints 202 (e.g. ITM contracts), and SOS SOA transaction error-handling logic 212. The ITM 107 may also include common error-handling logic, application-specific extensions to the common error handling logic, and versioning and transposition rules. The BODVRR 102 contains ITC and ITM documentation, including: design information, change history, change approval, change approver identification, XML, Unified Modeling Language (UML) scenarios, and deployment information. This is self-documenting.

[029] The definition of an ITC and the ITM that transports it separates business information content from technical implementation and routing information. The ITM contains flags for security Protection and Export Control that provide for optional processing to encrypt an ITM or to log the transaction for regulatory compliance purposes. ITM manages transfers of data exceeding the maximum size of a single XML document through the use of external references to data stores that contain ITC payload.

[030] FIG. 3 illustrates a schematic diagram of a BODVRR and gateway platform 300 according to an illustrative embodiment of the invention. ITMs, ITM contracts, and scenarios are loaded into the BODVRR 102 during development. The loading may occur via an SDS interface display (FIG. 5). Next, the non-populated or flat ITMs and ITM contracts are sent into a gateway 302 at runtime. Then, populated, predefined ITM messages are exchanged, based on the ITM contracts. An application 112 populates the ITMs via an API translator 307 and a BOD converter 304.

[031] Text Editors such as XMLSPY, POSEIDON, and MICROSOFT OFFICE WORD are types of Text Editing tools. Unlike the BODVRR, they do not support a storage methodology. They support the transposition from older to new versions of the ITMs, but not at runtime as BODVRR does. Similarly, ORACLE, MICROSOFT ACCESS, FOXPRO, etc., are all types of Database storage tools. While BODVRR uses a relational database technology, which is supported by the above tools, to store and manage the ITMs, these tools do not support an integrated editing and logical transposition of business objects as BODVRR does.

[032] Still further, XMLSPY, POSEIDON, MICROSOFT OFFICE WORD, MICROSOFT VISIO, etc., are all types of XML-based Graphical Editing tools. Unlike the BODVRR, they do not support a storage methodology. They support the transposition from older to new versions of the ITMs, but not at runtime. Additionally, XMLSPY, POSEIDON, MICROSOFT OFFICE WORD, MICROSOFT VISIO, etc., are all types of XML Validation tools. Unlike the BODVRR, they do not support a storage methodology. They support the validation or transposition from older to new versions of the ITMs, but not at runtime. Also, CVS, MICROSOFT SOURCESAFE, IBM CLEARCASE, etc., are all types of versioning tools. Unlike the BODVRR, they do not supply a runtime component that allows for ITM translations between versions. They also do not allow for the transposition from older to new versions of the ITMs. Embodiments of the present invention demonstrate the capabilities and overcome all the shortcomings of conventional systems in an integrated fashion.

[033] FIG. 4 illustrates an operational flow 400 performed in representing business transactions according to an illustrative embodiment of the invention. When reading the discussion of the routines presented herein, it should be appreciated that the logical operations of various embodiments of the present invention are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations illustrated in FIG. 4, and making up the embodiments of the present invention described herein are referred to variously as operations, structural devices, acts or modules. It will be recognized by one skilled in the art that these operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof without deviating from the spirit and scope of the present invention as recited within the claims set forth herein.

[034] The operational flow 400 begins at operation 402 where the BODVRR 102 determines business need responsibility. Next at operation 404, the BODVRR 102 specifies services associated with the responsibility. At operation 405 the BODVRR 102 designs ITMs for services. Then at operation 407, the BODVRR 102 determines ITM stewardship or ownership. The operational flow 400 then continues to operation 410.

[035] At operation 410 the BODVRR 102 identifies senders and receivers associated with a business transaction. Then at operation 412, the BODVRR 102 models data to be exchange. At operation 414 the BODVRR 102 performs an enterprise search and a gap analysis at operation 417. Next, at operation 420 the BODVRR 102 generates a detailed ITM design.

[036] At operation 422, the BODVRR 102 implements the ITM. Next at operation 424 the BODVRR 102 determines whether a National Institute of Standards (NIST) test has been made. The operational flow 400 then continues to operation 427 where ITMs are tested and certified. Next, at operation 428, the ITM is certified. At operation 430 the BODVRR 102 obtains ITM XML message certification and publishes the ITM at operation 432. The ITM is now ready for Enterprise production. BODVRR ITMs and ITCs must be reviewed and approved by the application integration administrator role before they will be used in production or included in the production extract.

[037] FIG. 5 illustrates a display 500 utilized for viewing inputs and output of a repository according to an illustrative embodiment of the invention. The display 500 illustrates input fields for functionality 502 to create a BOD. Similarly, the display 500 includes other access tabs, such as the search tab 504. It should be appreciated that the display 500 is utilized to load BOD data outside of runtime. The BODVRR 102 controls the input, revision, reporting, and release of ITC and ITM through functional roles that persist permission categories to all functions of the BODVRR.

[038] The BODVRR 102 also exports a selected subset of the ITC and ITM for use by an external interface program to verify the ITM is executing an approved ITC (e.g. the Validation capability. Additionally, the BODVRR administers the versioning of ITMs that are transmitted between applications. The business logic associated with an ITC can adapt between two different versions of the ITM when that conversion has been documented in the release of a new version. And BODVRR will provide the information where there is no valid contract between two different versions of an ITC and generate a standard error response that will be returned to the sending application.

[039] FIGS. 5a-5c illustrate a block diagram of an SOA 500 implemented by an SDS integration platform according to illustrative embodiments of the invention. The SOA 500, in FIG. 5a, includes a business layer 502, a gateway layer 504, and an application layer 505. The business layer 502 includes logical subsystems, modules, or components of each integrated system of systems, such as logical subsystems A1 507 and B1 510. Physical communication between the integrated systems occurs via the business layer 502 utilizing reusable business transactions such as a message 508.

[040] The gateway layer 504 includes gateways, such as gateways A2 512 and B2 514, which connect a logical subsystem of the business layer 502 with one or more applications of the application layer 505 such as applications A3 515 and B3 517. For instance the logical subsystem B1 510 is connected to the application B3 517 via the gateway 514. The gateway 514 regulates

communication to and from the application B3 517. Thus, communication from the application A3 515 to the application B3 517 is routed via the business layer 502 through the gateways A2 and B2 to facilitate flexible changes to the application layer 505 with minimal changes to the business layer 502. Additional details regarding facilitating changes to the application layer 505 will be described below with respect to FIG. 5b.

[041] FIG. 5b illustrates the SOA 500' when a change to the application B3 517 occurs. When the application B3 517 is replaced by a new application B3 522, a new gateway B2 520 is also generated to regulate communication to and from the new application B3 522. For instance, the new gateway B2 translates, reformats, repackages, and/or filters source data in the form of a message such as the message 508 to be delivered to the new application B3. Similarly, when the new application B3 produces a result, the new gateway B2 translates, reformats, repackages, and/or filters the result to be presented in the form of a defined message or reusable business transaction. Applications can be replaced, without affecting the overall SDS system of systems, by selecting a new application, such as the new B3, implementing a new gateway, such as the new B2, and attaching the new gateway to a messaging or application adapter.

[042] FIG. 5c illustrates the SOA 500" according to another embodiment of the present invention. A single functional unit or logical subsystem, such as the logical subsystem B1 510', may reside as a single component implemented using a single application within the SOA 500". However, the single component may also be implemented using a combination of applications, such as the applications 517a-517c, where each application is mediated by a gateway, such as gateways 514a-514c.

[043] Thus, the present invention is presently embodied as methods, systems, computer program products or computer readable mediums encoding computer programs for representing a business transaction.

[044] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

WHAT IS CLAIMED IS:

1. A computer-implemented method for sustaining a fleet of configuration-controlled assets, the method comprising:

integrating a plurality of previously separate systems with reusable business transactions associated with managing a fleet of assets wherein functional capabilities of the previously separate systems are defined by the reusable business transactions;

receiving operational data associated with at least one asset of the fleet;

integrating said data per customer requirements;

packaging and distributing said data to the plurality of systems; and

installing and distributing said data to one or more applications of the plurality of systems utilizing at least one of the reusable business transactions.

2. The method of claim 1, further comprising:

receiving operations data regarding the at least one vehicle in response to the vehicle being in operation;

evaluating and determining maintenance needs of the at least one vehicle to create a work order;

determining whether parts are available to fulfill the work order; and

determining an optimal schedule to perform the work order.

3. The method of claim 2, further comprising determining whether a trained resource is available to perform the work order.

4. The method of claim 2, further comprising receiving performance of the work order and recording the work performed.

5. The method of claim 2, further comprising tracking and reporting one or more anomalies detectable via one or more of the plurality of systems.

6. The method of claim 2, further comprising:

aggregating metrics data associated with sustaining the fleet; and

calculating one or more key performance indicators (KPIs) based on the metrics data collected wherein the KPIs are identified by the customer.

7. The method of claim 6, further comprising:

forwarding at least one of the metrics data or the KPIs for engineering system analysis and feedback; and

analyzing the metrics data or KPIs for reliability, maintainability, and performance trends related to the fleet and its assets.

8. The method of claim 7, further comprising:

forwarding at least one of the metrics data or the KPIs for logistical system analysis and feedback; and

analyzing the metrics data or KPIs for logistical reliability, logistical maintainability, and logistical performance trends related to the fleet and its assets.

9. The method of claim 1, further comprising managing users of the plurality of previously separate systems via a central authoritative logon for the plurality of systems.

10. The method of claim 9, wherein managing users via the central authoritative logon comprises:

receiving logon inputs from one or more users;

identifying and authenticating the users; and

issuing a role-based credential for each user to be used during a user session accessing one or more of the plurality of systems.

11. The method of claim 1, wherein the plurality of systems comprise at least two of a vehicle mission planning system, a supply chain management system, a training

management system, a vehicle configuration state database, an optimized resource planning service, a technical publications system, a maintenance management system, and an integrated vehicle health management system.

12. A computer-implemented system for integrating a plurality of applications, the system comprising:

an integration manager operative to manage the insertion and removal of applications to and from the plurality of applications;

an integration gateway operative to interface reusable business transactions and the plurality of applications;

one or more application adapters interfacing the plurality of applications with the integration gateway; and

an intelligent transaction manager extract operative to extract and install reusable business transactions.

13. The system of claim 12, further comprising a single sign-on authentication module operative to:

manage users of the system via a central authoritative logon point for the plurality of applications;

identify and authenticate a user; and

issue role-based credentials for the user to use during an engagement comprising a time from logon through a time of logout with a maximum time period.

14. The system of claim 12, wherein the reusable business transactions utilize previously latent features in one or more of the plurality of applications.

15. The system of claim 12, wherein the reusable business transactions are documented in XML and wherein storage and representation of the reusable business transactions are enabled to extend to alternative technologies.

16. The system of claim 12, further comprising an encryption module operative to verify the reusable business transactions with an encrypted relational database against at least one of a primary real-time repository or a runtime extract of the repository.

17. The system of claim 12, further comprising a system integration business logic service operative to:

instantiate a service-oriented architecture (SOA);

permit interconnection of previously separate applications through the reusable business transactions; and

prohibit direct use of external interfaces controlled by the system.

18. The system of claim 12, further operative to implement modular advanced services for the plurality of applications, therein adding capability to the system via a utility application that leverages the reusable business transactions when a customer needs capability beyond existing capability of the application.

19. The system of claim 12, further operative to implement an integrated transmission test capability to capture transmitted reusable business transactions into an audit log.

20. A computer program product comprising a computer-readable medium having control logic stored therein for causing a computer to sustaining a fleet of configuration-controlled assets, the control logic comprising computer-readable program code for causing the computer to:

integrate a plurality of previously separate systems with reusable business transactions associated with managing a fleet of assets;

receive operational data associated with at least one vehicle of the fleet;

integrate the operational data per customer requirements;

package and distribute the operational data to the plurality of systems; and

install and distribute the operational data to one or more applications of the plurality of systems utilizing at least one of the reusable business transactions.

21. A system for sustaining a fleet of vehicles, the system comprising:

a plurality of applications associated with the management of a fleet of mobile platforms;

a plurality of logical subsystems, each of which associated with a respective one of the plurality of applications; and

a plurality of gateways, each of which connects one of the plurality of applications to a respective one of the plurality of logical subsystems and converts the data flow between the applications and the subsystems.

22. The system of claim 21, wherein the plurality of the logical subsystems utilize a common format defined by a set of messages.

23. A method for sustaining a fleet of vehicles, the method comprising:

integrating a plurality of modules associated with the management of a fleet of mobile platforms; and

creating a predetermined set of messages for allowing the plurality of the modules to communicate with each other.

24. The method of claim 23, wherein the plurality of the modules are integrated vehicle health management system, maintenance management system, supply chain management, training management system, and resource optimization scheduler.

25. The method of claim 23, further comprising the step of monitoring the performance of the plurality of separate systems.

26. The method of claim 23, further comprising the step of monitoring the anomalies of the plurality of separate systems.

27. The method of claim 23, further comprising the steps of analyzing the anomalies of the plurality of separate systems and providing a recommendation.

28. A method for managing the insertion and removal of previously separate domain applications that have been integrated with one another, said method comprising:

configuring an extended application interface to connect the domain applications together;

providing a gateway operative to interface reusable transactions with the domain applications;

operating a transaction manager to extract and install the reusable transactions; and

implementing an application adapter for each new domain application to be added, with the application adapter configured to interface the new domain application to the extended application interface.

29. The method of claim 28 wherein configuring an extended application interface to connect the domain applications together comprises combining message-oriented middleware and service oriented architecture with modular application gateway adapters and standard formatted messages.

30. The method of claim 28 further comprising connecting existing systems with logic contained in modular application gateway connectors through verified transactions with anti-spoofing capabilities.