Model Driven and Service Oriented Enterprise Integration---The Method, Framework and Platform

Shuangxi Huang, Yushun Fan

Department of Automation, Tsinghua University, 100084 Beijing, P.R. China
{huangsx, fanyus}@tsinghua.edu.cn

Abstract

SOA and MDA have emerged as a major evolutionary step in enterprise integration. Significant advances have been seen in SOA and MDA. But the integration of SOA and MDA is still lacking. In this paper, the methodology of model driven and service oriented enterprise integration is proposed. It uses MDA as the philosophy of system development and SOA as the infrastructure of system implementation. After that, the technological framework of the methodology is presented to support the realization of integration in both model and service level. The framework comprises the meta model, the model driven development frame, and service orient executing environment. Finally, the functionalities of the SOA and MDA based enterprise integration platform is introduced, which provides the integrated development and operational environment for enterprise integration.

1. Introduction

Enterprise Integration (EI) has long been foreseen as the solution to a wide range of problems, enabling companies to reduce time to market, to improve quality, to increase supply chain efficiency and even to understand customers better. To date, this vision has not achieved marked success. The permanent changing manufacturing world makes enterprise integration become a no ending task. Nowadays, the business situation is characterized as globalization, customization, digitization, virtualization, and agility. To tackle with the challenges, new technologies, methods and strategies are needed. More recently, SOA and MDA have emerged as a major evolutionary step in EI areas.

Service-Oriented Architecture (SOA) is a software architecture consisting of a collection of loosely coupled services that communicate with each other through open-standard interfaces. The World Wide Web Consortium (W3C) refers to SOA as ‘‘a set of components which can be invoked, and whose interface descriptions can be published and discovered’’ [1].

Model-Driven Architecture (MDA) is proposed by the Object Management Group (OMG) as a reference to achieve wide integration of enterprise models and software applications [2]. MDA is a best choice to address how SOA should be designed, developed and integrated.

In this paper, based on SOA and MDA, an integrated solution for enterprise integration is proposed which supports the full-scale integration of enterprises, from conceptual integration in model level to technological integration in IT level. It uses MDA as the methodology of system development to achieve integration at model level including all the enterprise systems' life-cycle models. As well as SOA is used for the implementation of model and for the construction of integration platform.

This paper is structured as follows. The related work is provided in the next section. In Section 3, the methodology of model driven and service oriented enterprise integration is proposed. Section 4 gives the technological framework to support the implementation of the methodology. In section 5, the functionalities of SOA and MDA based enterprise integration platform is introduced, which provides the integrated development and operation environment for enterprise integration. Finally, conclusions are made in section 6.

2. Related work

In the field of enterprise integration, according to the level of integration， the research work can be organized as six classifications which are organization integration, process integration, data integration, application integration, service integration, and semantic integration.

- Organization integration is mainly concerned with networked organizations and value models. Important scientific contributions to organization integration have come from transaction cost theory, organizational theory, coordination theory, and business networks. Based on the review of different approaches to integration, Christine Legner derives the relevant artifacts and proposed a comprehensive business integration framework to address the interactions between business strategy, organizational design and information system design [3].
- Process integration mainly concerns integration at the business level, i.e. business process integration (BPI) and cross-organizational workflow management. It enables an enterprise to respond with flexibility and speed to changing business conditions by integrating its business processes end to end across the company and with key partners. Some industry standards, such as RosettaNet, ebXML, Common Business Library (CBL), Electronic Data Interchange (EDI), Open Buying on the Internet (OBI), and cXML (Commerce XML), provide guidelines to achieve integration among business processes of individual organizations. The standards define common ontology, syntax for the message exchange, and interactions across organization boundaries [4].
- Data integration refers to make work together between different data models (hierarchical, relational, etc.). Data integration research relates to the syntactic and semantic differences of information to be exchanged as well as the expressivity of the information. Three kinds of methods are used in data integration, which are integrated, unified, and federated methods. Some technologies, such as enterprise model, XML, UEML, ontology, etc. can be used in data integration [5].
- Application integration concerns integration of applications on heterogeneous platforms as well as access to shared data by the various remote applications. Distributed processing environments, common services for the execution environment, application program interfaces (API's), and standard data exchange formats are necessary at this level to build integrated systems. Application integration started in the mid 1980's and is still on-going with very active work concerning STEP, EDI, HTML, XIVIL, or ebXML for the exchange of common shared data, development of common services for open systems around the web (web services), and integration platforms for interoperable applications in distributed environments (e.g. OSF/DCE, OMG/CORBA, and more recently J2EE or .NET) [6].
- Service integration is concerned with identifying, composing and making function together by solving the syntactic and semantic differences of services. Key to developing service integration is the service-oriented architecture (SOA). Currently, the SOA provides the basic operations necessary to describe, publish, find and invoke services [7]. Except SOA, some other research efforts have been made to facilitate Service integration, such as service modeling, service composition and choreography, etc..
- Semantic integration crosses all the levels of EI which enable the common or equivalent semantic between enterprises so that related entities can communicate and understand each other. Ontology and Semantic Web are the two core technologies of semantic integration. More recently, the Semantic Web Services emerges aiming to combine concepts of the Semantic Web with web services. The WSMO (Web Service Modeling Ontology) and OWL-S can be used to describe the concept of service, as well WSPO (Web Service Process Ontology) provides a standard template for service process description [8].

Over the last decade numerous efforts have been made in the area of enterprise integration. Among them, SOA and MDA are now considered as mainstream and have been gaining momentum over the last few years.

As an architectural style for distributed systems, SOA provides a broad insight to tackle with enterprise integration. The most frequently used method for implementing SOA is web services. SOA allows enterprises to dynamically publish, discover and aggregate a range of Web services through the Internet and offers mechanisms of flexibility and integration that allow different technologies to be dynamically integrated, independently of the system's platform in use. Web services are not single but a collection of a variety of technologies offering services' description, registration, publication and search functionalities [9].

Although providing a significant contribution, the SOA alone is not yet the answer to achieve EI. Some crucial problems still related with the implementation of EI in SOA environment, such as development methodology, the dynamic combination and usage of services, the consistency between business and information system.

Model-Driven Architecture (MDA) is a best choice to address how SOA and Web services should be designed, developed and integrated in order to achieve integration. MDA provides specifications for an open architecture appropriate for the integration of systems at different levels of abstraction and through the entire information systems' life-cycle. The MDA comprises three main layers: Computation-Independent Model (CIM), Platform-Independent Model (PIM), Platform Specific Model (PSM) [10]. MDA lies in separating the enterprise model from the technology infrastructure, making a clear division between the business functions and the implementation details. When a requirement changes the business behavior, this change is reflected in the abstract level (i.e., PIM) and it will be directly mapped to the system through the PSM. Hence, the enterprise model is transformed to be applied to different technology deployments like J2EE or .NET.

3. Methodology for model driven and service oriented enterprise integration

Traditional enterprise systems suffer from lack of integration due to the large number of system's heterogeneity and the adaptivity of architecture. Today, the emergence of SOA and MDA provides an adequate approach to enterprise integration. MDA lies in separating the enterprise model from the technology infrastructure, making a clear division between the business functions and the implementation details. With this architecture the modeling effort is safeguarded independently of the specific implementation platform. The transformations to the PSM are the mechanism to assure the reusability of the abstract model towards the implementation in the target platform, and in this way to assure the integration of the implemented systems with the other also adopting the same PIM.

On the other hand, the service-oriented architecture (SOA) is a method for building software applications that use services registered and available in a network. A service implements enterprise's functionalities that can then be used by clients in different applications, offering a framework for a better integration between the software applications that use the services.

The MDA and SOA represent the current best-practice in how to develop new software and how to recondition legacy applications in order to design integrated systems.

In this paper, a comprehensive methodology for model driven and service oriented enterprise integration is developed. It uses MDA as the methodology of system development to achieve integration at model level including all the enterprise systems' life-cycle models. As well as SOA is used for the implementation of model and for the construction of integration platform.

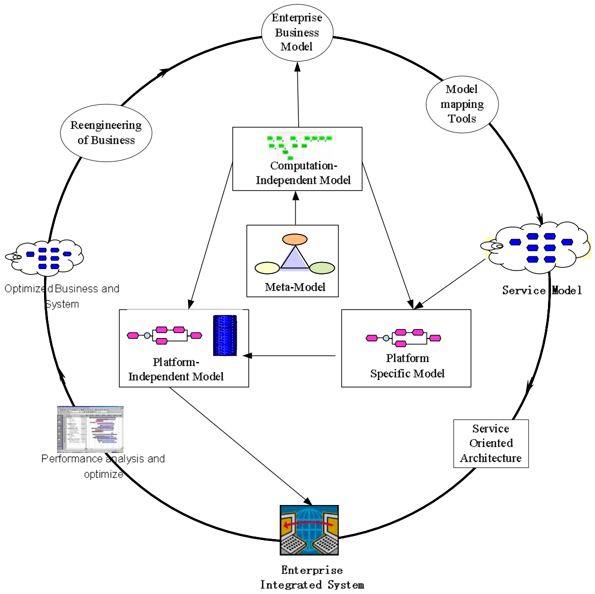
Enterprise Business Model
Model mapping Tools
Reengineering of Business
Computation-Independent Model
Meta-Model
Optimized Business and System
Service Model
Platform-Independent Model
Platform Specific Model
Performance analysis and optimize
Service Oriented Architecture
Enterprise Integrated System

Fig.1. Methodology of model driven and service oriented enterprise integration

Figure 1 shows the methodology of model driven and service oriented enterprise integration. Driven by the requirements of business integration, the enterprise integration model can be constructed based on integration meta-model. The enterprise integration model is a Computational Independent Model (CIM) that describing the business requirements of integration, including integration roles, processes, information and services, etc. In terms of the concepts of services in SOA, the computational independent integration model can be transformed to a platform independent model. Services are conceptual entities that include business and realization of business. Therefore, the computational independent integration model can be transformed into a platform independent service model, which can be realized and supported by computer systems. There are many technologies that can be utilized to implement the functionalities of services and platforms for SOA, such as J2EE, .NET, CORBA, etc. According to existing systems and platforms, different enterprises can choose different implemental technologies to transform platform independent service model into platform specific model.

When the PSM is developed, under the support of services infrastructure, using the service bus, through the composition and choreography of service, the SOA and MDA based enterprise integration system can be established.

During the running time of the system, the performance of the integrated system is monitored and analyzed. The architecture, services and the business processes can be optimized. The optimized integration model will be re-defined and released by integration modeling tool, then a new cycle of development, deployment, operation, optimization is started.

4. Framework for SOA and MDA based enterprise integration

In this paper, the technological framework to support the implementation of the methodology is presented, which can realize the integration both in model and service levels. Figure 2 shows the framework of SOA and MDA based enterprise integration. The framework is composed of three levels: Meta model level, modeling level, and service level.

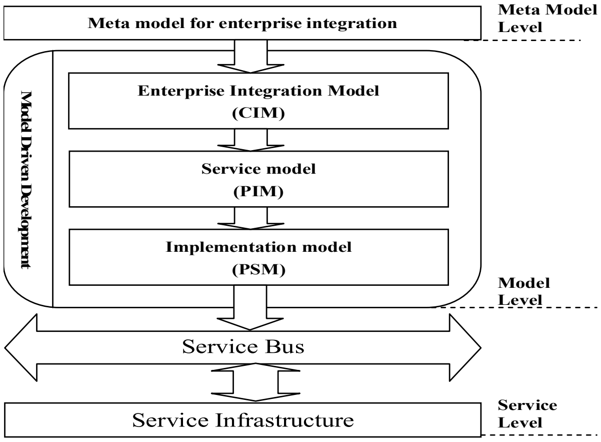

Fig. 2. Framework for SOA and MDA based enterprise integration

In meta model level, the meta model of enterprise integration is defined aiming at describing fundamental constructs for computation independent modeling language that can be used to model the integration from business perspective. Five basic elements are defined in the meta model. They are organization, process, data, system and service. Moreover, for the convenience of integration modeling, some other concepts are defined in the meta model, which are IntegrationObjects, IntegrationInstance, Role, RoleInstance, and Rule. The IntegrationObjects specifies the involved roles and their responsibilities. Five kinds of roles are defined in meta model: Organization, Process, Data, System and Service. IntegrationInstance specifies the application of an IntegrationObjects in a specific context and includes the RoleInstances. RoleInstance refer to the Role binding to operational objects. Furthermore, IntegrationObjects should act in accordance with Rule that specifies the constraints that define how the involved Roles interact. The meta model of enterprise integration is presented in Figure 3.

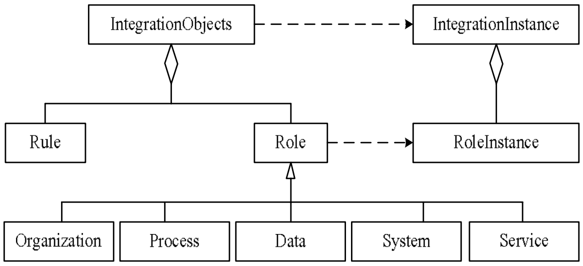

Fig. 3. Meta model of enterprise integration

The modeling level comprises three main models: enterprie integration model (CIM), service model (PIM), implementation model (PSM).

1) Enterprise integration model is a computation independent model (CIM) which is a business oriented representation of an enterprise integration system from the computation independent viewpoint. The enterprise integration model focuses on the business and business environment in which a system will be used, abstracting from the technical details of the structure of the implementation system. It describes the business context and business requirements for enterprise integration.
2) Service model is a visual PIM which specifies services in a technology independent manner. It represents an integrated view of the SOA in which different components can be deployed on different execution platforms. The service model helps us to align relevant aspects of enterprise and technical IT models, such as process, organization and products models. This model allows us to raise the abstraction level at which we can talk about and reason on the architecture we design.
3) Implementation model is also a visual model which IT developers can further refine by adding platform-specific modeling constructs such as deployment properties. The service model can transfer into different implementation models by underlying different platforms and protocols, e.g., CORBA and Web services, or the different implementations of SOAP.

The service level includes service bus and service infrastructure. The service bus provides the necessary communication infrastructure required to deploy a distributed system. Service infrastructure enables integration between services through the composition, mediation, matchmaking and transformation of services.

The integration of SOA and MDA fits the gap between concept (model) and technology (implementation) and maintains the consistency of business requirement and IT implementation by using model driven and service oriented approach.

5. SOA and MDA based Enterprise Integration Platform

This section describes the functionalities of SOA and MDA based Enterprise Integration Platform (SMEIP). The SMEIP is a development and operational platform for the enterprise integration. It provides an integrated development and execution environment based on SOA and MDA. The SMEIP is composed of the following eight parts [Figure 4]:

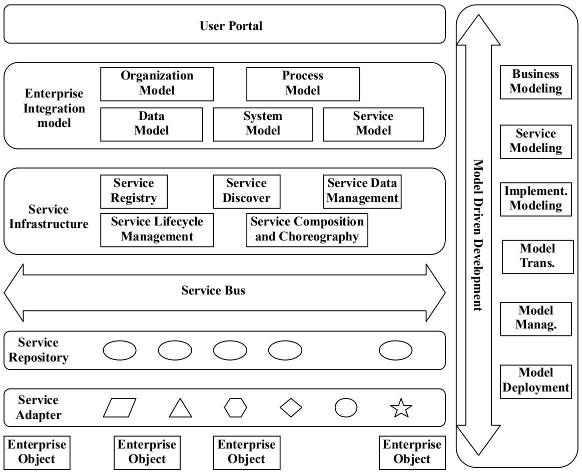

Fig.4. Architecture of SOA and MDA based enterprise integration platform

1) Enterprise Object: It refers to the elements involved in integration. All the elements defined in enterprise integration model can be Enterprise Objects of SMEIP, such as enterprise systems (ERP, PDM, SCM, etc.), business processes, organization units, or data. Before integrating in SMEIP, the Enterprise Object must be wrapped by Service Adapter.
2) Service Adapter: It converts the access interface of Enterprise Object into standard service interface. The adapter works as an intermediary between service and Enterprise Object. It models data and functionality present in Enterprise Object and exposes a service interface from which functionality can be access.
3) Service Repository: It stores models and interface descriptions in a central location that is accessible to Enterprise Objects. A repository will permit searching for models and interface descriptions. A repository will map logical service identifiers with physical addresses.
4) Service Bus: It is a standards-based communication layer in a service-oriented architecture (SOA) that enables services to be used across multiple communication protocols. Service Bus provides an intermediation approach which abstracts the interaction details between a service consumer and one or more service producers. It simplifies service deployment and management and promotes service reuse in a heterogeneous environment.
5) Service Infrastructure: It is a unified approach to services management that combines the capabilities of Enterprise Service Bus, business process management, and services. Service Infrastructure creates a fabric of services for constructing and deploying new composite business applications. The main functions of Service Infrastructure include service registry, service discover, service data management, service composition and choreography, and service lifecycle management.
6) Enterprise Integration Model: It defines the business model of integration. Five kinds of business model of integration are defined in organization, process, data, system, and service levels. Through the composition of different business models, the actual scenario model can be established. The eligible services can be matched, composed and executed through the mapping of CIM to PIM and PIM to PSM.
7) User Portal: It is the front end of platform, supporting dynamic workplaces and working environments tailored to purpose. It generates a dynamic and configurable user interface which makes the user access platform functions and the back end applications transparently and obtaining the right information at the right time.
8) Model Driven Development Framework: It crosses all levels of SMEIP architecture, providing various MDA tools for the development of EI system including business modeling, service modeling, implementation modeling, model transformation, model management, and model deployment.

6. Conclusions

This paper proposes a SOA and MDA based enterprise integration solution which includes methodology, technological framework, and integration platform. The combination of SOA and MDA can enhance the agility, flexibility, and robustness of integration by using model driven and service oriented approach. MDA gives the opportunity to bring the services definition to a higher level of abstraction, due to the more formal and accurate platform-independent specification of the services requirements and design. Afterwards, to have the services implemented in a specific platform, a suitable transformation is executed. The high-level models can be transformed to service implementation. This way, services and SOA can be created decoupled from the lower level platforms, infrastructures and implementations, opening the way to improved integration. With this complementary joint approach, we take the most from SOA and MDA, stimulating the reusability and improving the integration in different levels.

Acknowledgments. The research of the paper is supported by the National Natural Science Foundation projects of China under Grant No. 60504030, the National High Technology Research and Development Program of China under Grant No. 2006AA04Z166, and the European Commission FP6 project ImportNET.

7. References

[1]. W3C, World Wide Web Consortium(2006), http://www.w3c.org.

[2]. Object Management Group (2006), available at: http://www.omg.org.

[3]. Christine Legner, Kristin Wende. Towards an Excellence Framework for Business Interoperability. 2006, available at: http://www.bledconference.org

[4]. Mehmet Sayal, Fabio Casati, Umesh Dayal, Ming-Chien Shan: Integrating Workflow Management Systems with Business-to-Business Interaction Standards, Proceedings of the 18th International Conference on Data Engineering(2002) 287-296.

[5]. F.B. Vernadat, Enterprise Modelling and Integration: Principles and Application, Chapman and Hall, London, 1996

[6].F.B. Vernadat: Enterprise Modeling and Integration(EMI):Current Status and Research Perspectives. Annual Reviews in Control. 26 (2002) 15-25

[7]. Booth, D., Haas, H., McCabe, F., et al.: Web Services Architecture, W3C Working Group Note (2004), available at: http://www.w3c.org/ws-arch/

[8]. R. Lara, D. Roman, A. Polleres, and D. Fensel. A Conceptual Comparison of WSMO and OWL-S. In L.-J. Zhang and M. Jeckle, editors, European Conference on Web Services ECOWS 2004,. Springer-Verlag.(2004) 254–269.

[9]. Arne-Jørgen Berre, Axel Hahn, David Akehurst, etc.. State-of-the art for Interoperability architecture approaches. Deliverable D9.1 of InterOP (IST 508 011). 2006, available at: http:// www.interop-noe.org

[10]. Object Management Group (2006): Model Driven Architecture, MDA Guide Version 1.0.1 (2003), available at: http://www.omg.org/mda.