Integrated Enterprise Modeling Framework for Developing Consistent Distributed Systems

Yu Zhao, Yushun Fan
Department of Automation, Tsinghua University, Beijing 100084, China
Zhaoyu00@mails.tsinghua.edu.cn, fan@cims.tsinghua.edu.cn

Abstract—Although there are many existing and standard methods, such as RM-ODP and MDA, to develop a single distributed system to assure it consistent with the business requirements, the different enterprise-wide systems of one same enterprise might not be consistent with each other themselves because of the inconsistent business requirement model for each other. Enterprise model could describe the common business architecture of the enterprise and could be the common resource of the requirements for every system. In order to solve the integration and consistency problem, an integrated enterprise modeling framework for developing the enterprise-wide distributed systems is proposed, and the process and rules for extracting the system architecture from business model are also discussed.

Keywords—distributed system, enterprise model, enterprise modeling, requirement engineering, RM-ODP, system architecture

INTRODUCTION

In the present-day enterprises, information systems play more and more important role in strengthening the competency. As a result, more and more enterprise-wide distributed systems need to be developed for various domains, such as ERP for resource planning, SCM for supply chain management, PDM for product management and so on. For a single distributed system, there have some standard methods and approaches for supporting its development, such as

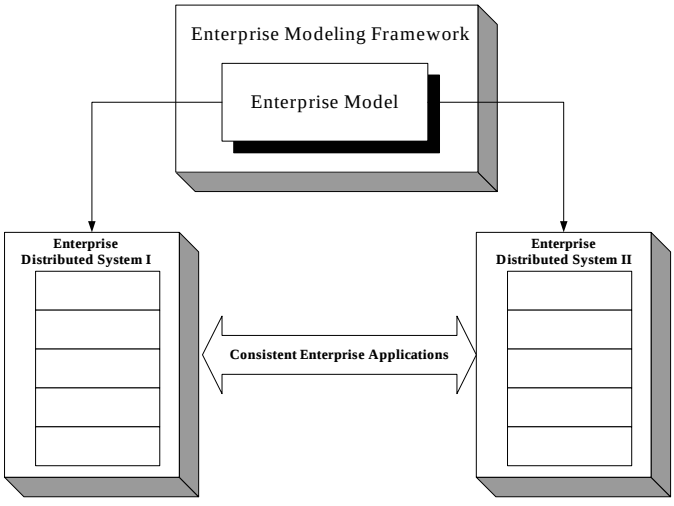

Fig. 1. Developing the consistent enterprise applications based on the common enterprise model.

RM-ODP (Reference Model for Open Distributed Processing) [1] and MDA (Model Driven

Architecture) [2]. The final purpose of most of the methods is to guarantee the consistency between the business requirements and the system design. Although this idea is quite right for system development, it is only suit with the development of a single system. Along the RM-ODP or MDA, each system has its own business requirements model, and these requirements models might be created by different business analyzer or at different time for one same enterprise. However, for an enterprise, the different systems usually overlap each other in some business domains or share the same business data for different purposes. Thus the situation is more and more severe for most enterprises that all the enterprise-wide systems are inconsistent with each other and unable to be integrated together easily.

Enterprise model and enterprise modeling technology have been proposed for the purpose of solving the enterprise integration problems [3]. Enterprise model describes the business architecture of the enterprise, and is regarded as the common fundamental data of the organization. All applications have to be consistent with the enterprise model, and all modifications of the systems must root in the modifications of the enterprise model (Figure 1) [4]. But up to this time the focus of enterprise modeling research is still on the enterprise modeling framework and modeling methods, while how to assure the consistency between the system and the enterprise model, which is the key problem for making enterprise integration operational, has been seldom discussed on. The big gap between the business and the systems still exists.

So in this article, we focus on researching how to extract system basic architecture for enterprise-wide distributed system development from enterprise model, and propose an integrated enterprise modeling framework for distributed systems and the rules for extracting system model from business model under this framework as results. RM-ODP and MDA, which are purely software development methodology, are introduced in Section 1. Then, some concepts of enterprise model and enterprise modeling are described in Section 2. The enterprise modeling framework integrating the business aspect with the software aspect is proposed in Section 3. In Section 4, the mapping process from business model to system model is analyzed.

1 RM-ODP AND MODEL DRIVEN ARCHITECTURE

The RM-ODP (Reference Model for Open Distributed Processing) is a standard developed by ISO (International Standard Organization) and focuses on the specification of distributed systems. It uses simultaneous multi-viewpoints on a distributed system to describe characteristics of that system from various aspects in a standard way. The viewpoints are: enterprise viewpoint – describes what the system needs to do, capture the business and management requirements, and decides the strategies for the system design with enterprise viewpoint model; information viewpoint – describes the information infrastructure, information flow and the constraints for information operation in the system; computational viewpoint – describes the detailed operations in system, i.e., the computation of the process to modify information; engineering viewpoint – describes the engineering resources supporting the distributed processing; technology viewpoint - describes the practical elements needed by the system. In essence, the static viewpoint structure of RM-ODP is not flexible for variety development cases, but we should take notice of the core thought of RM-ODP that is to model systems at different levels and to separate the system structure from the computation and data in order to clarify the business requirements.

In recent years, OMG (Object Management Group) emphasizes the similar ideas with RM-ODP for the software development. And it has transferred its stress from the middleware standard -

CORBA to the system development process standard – MDA. According to MDA, model is not only used to describe system, but also to be the core of the system development. All models for system description should be divided by different development levels, and each model should be only to describe the system from some aspects for one development level. In MDA, the models are divided into three modeling levels – CIM (Computation Independent Model) Layer, PIM (Platform Independent Model) and PSM (Platform Specific Model) layer. A CIM is a view of a system from the computation independent viewpoint, and it does not show details of the structure of systems. A CIM might consist of models from enterprise viewpoint and information viewpoint of RM-OPD. A PIM is a view of system from the platform independent viewpoint, describes the system, but does not show details of its use of its platform. A PIM might consist of models from enterprise, information and computational viewpoint of RM-ODP. A PSM is a view of system from the platform specific viewpoint, and it combines the specifications in the PIM with the details that specify how that system uses a particular type of platform.

2 Enterprise Model and Enterprise Modeling

Enterprise model describes one or several aspects of enterprise through abstraction, in order to make people understand the enterprise. The scope and content of an enterprise model are adapted by the purpose of modeling. Enterprise modeling is the process of building the enterprise model, and is concerned with externalizing and structuring knowledge about various aspects of an enterprise, such as functional, information, organization, economic or resources aspects. Over the last decade, enterprise modeling has received considerable attention and many modeling frameworks (such as CIMOSA, ARIS, IEM, GRAI/GIM), methods, languages and tools (such as ARIS, FirstSTEP, MEGA, Metis) are developed. The three major thrusts for the rapid growth of enterprise modeling techniques have been: Enterprise Engineering (i.e. systematic and shortened development cycle engineering of a business entity, including Business Process Reengineering or BPR techniques), Enterprise Integration (i.e. increasing synergy and interoperation among people, systems and applications throughout the enterprise, including integration in manufacturing or CIM), and workflow management (WfM) dealing with automation of paper and document flows as well as control of business processes [5].

3 Integrated Enterprise Modeling Framework for Developing Distributed Systems

Enterprise modeling frameworks and methods vary with the purpose of modeling. We proposed an integrated enterprise modeling framework here for developing enterprise-wide distributed systems. The enterprise model built under this framework can be used to extract the system architecture, and pass the exact business requirements, which are reflected in the enterprise model, to the systems. In order to link the business with systems, this modeling framework is made up of five core views located at two levels: the business process view, the organization view, the resource view at business level; and the function view, the information view at system level. (Additional views are added for some special or detailed purposes such as economic view for economical evaluation and network view for network deployment.) These views are integrated together by the citation relationship among the views from the same level and the mapping relationship among the views from the different levels. This framework is illustrated by Figure 2. And, it is proposed based on the EMITS framework for enterprise informatization total solution. [6]

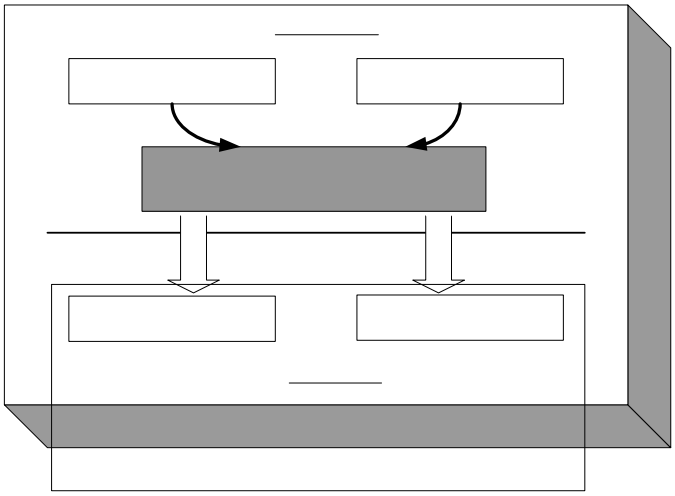

Fig. 2. integrated enterprise modeling framework for developing distributed systems

The business process view describes the work flow of enterprise through defining the activities which constitute the process and the logical relations among these activities. The organization view describes the organization structure of the enterprise for executing the business processes, including the organization tree, team, faculty, role and authority. The resource view defines the resource entity, resource type, resource pool and resource classification tree covering all the equipment resource, manpower resource and information resource supporting the execution of business processes.

The business process view is the core model of the business level under the integrated modeling framework. The relationships among the three views at the business level are represented by citations. Some contents in a view come from another view. Citation relationships are classified into three types: the organization cited by the business process; the resource cited by the business process; and the organization cited by the resource. It should be defined that which department unit and which role are responsible for the execution of each activity. And these contents defined in the business process view cite from the organization view. On the other hand, it should also be defined that who are the executors, which applications are used for the support, and which equipments are operated during the execution for each activity. And these contents also in the business process view come from the resource view. Since each resource unit has its owner, for example, some computers and software belong to Department A, and those employees are employed by Department B, we should define the owner information for resource units in the resource view which cite from the organization view.

In fact, models at the business level have two versions during the development of the enterprise systems. Before the design of the systems, the business model describes the active status of the enterprise without the systems, and we can call this version the As-is model. While the enterprise systems has been produced and implemented for the enterprise business, the business model should be modified to reflect the new status, and the model changed into the To-be model. The new systems supplement into the resource model. Some business activities perhaps unite into one single activity or modify their execution properties because of the new systems. And the organization structure may be modified according to the distribution of the new systems. So the As-is model represents the requirements of systems while the To-be model reflects the results of the systems' implementation.

The system level, another level of the integrated enterprise modeling framework, is for the

modeling of the enterprise system architecture according to the business models. In fact, the purpose of system level is similar to the computation independent and platform independent level of the MDA. The system models describe only the structure of the system components, the relationships, the information/data flow, and the information/data structure among these components. The models do not concern the detailed realization of these components and the relationships among them, and do not concern the final physical form of these information/data structure, either. At the system level, there are two core views supporting for the modeling of system named function view and information view, respectively. Function view describes the system's functional structure. Function model is made up of function units. Function units would be mapped into the sub-systems, components and modules of the system to realize the required enterprise business or management process. In the words of software engineering, the function view is the description of the software architecture, which has the same function as the enterprise viewpoint of RM-ODP. Information view sets up the semantic and logical structure of information extracted from the business process, and describes the data relationships and constraints, too. The content and use of information view equals to the information viewpoint of RM-ODP.

Thus, the process of the extracting the system architecture model, the enterprise and information viewpoints of RM-ODP, from enterprise model is the process of mapping from business level models to system level models under the integrated enterprise modeling framework. This extracting or mapping rules will be discussed in Section 4.

4 RULES FOR EXTRACTING SYSTEM ARCHITECTURE FROM BUSINESS

When the business level's models have been built, the enterprise managers, business analyzer or decision maker of company will diagnose and optimize the business model from their business aspects by many means, such as the benchmarking analysis with business reference models, and using varieties of emulator or mathematics tools to find the problems, errors or bottle-necks of the active business processes. According to the analysis results, correlative business processes, resource deployment and organization structure can be optimized to assure that the information system to be designed based on this business model is reliable at the aspect of the business at least. And they also proposed that which business process should be executed by information systems (or more urgently) to improve the business, as the results of analysis.

According to the optimized business model and the decision made by the business analyzer, the system architecture designer begins to take out and organize the business function units and functional structure of information system from the business process models by describing the function each business process or activity is operated to realize. The relations between business activities/processes and system functional units/components are that: one business activity/process embodies one functional unit; while one same functional unit could be realized by several different activities/processes. Using the OO semantics, a business activity/process is an instance of a corresponding functional unit. Thus the mapping from business process model to function model is a process of inducing some individual classes – functional units from instances – business activities. During the converting process, relationship among these activities and the information flow described in the business model are also mapped into the interaction between function units and the information flow of the whole information system, respectively.

Following the above mapping rules, the business process model is enough for the ordinary system architecture design. But for the distributed system specially, because of the characteristics of

distribution, the organization model at business level will play an important role. In nature, the requirements for distribution of system source from the status of distributed organizations. So after the function units have been induced from business processes, we should divide these functional units into several groups according to the organization model. For example, three business activities named A1, A2 and A3 correspond to functional units F1, F2 and F3, respectively. A1, A2 is executed in the Department D1, while the Department D2, located away from D1, is responsible for the execution of A3. In this case, A1 and A2 should be bound together to compose a sub-system S1, relative to the whole distributed system being developed, while A3 should belong to another sub-system S2 which is distributed from S1. However, there exists another problem when the system designer designs the distribution architecture of the system. Since one functional unit is usually realized by several activities which might be executed by different departments for each, some rules have to be established to deal with this case. There are two ways. We can copy such the units to several copies according to the number of departments who are responsible for this function. Thus these copies can be divided into different sub-systems in the same way as the other units which are discussed above. This is a simple solution, but it might result in the size, redundancy and complexity of the whole systems. Another way to cope with it is that the system designer can report this situation back to the business managers or analyzers, and suggest them to adjust the organization structure and deployment out of the concern for the organization efficiency because several different departments are doing the same things. And then the departments might be combined or a new department special for one function might be established. According to the new distribution of organizations, the system designer could rethink the system architecture. Obviously, this solution might result in the expensive cost for the organization adjustment. After all, the selection from these two solutions should be decided on the basis of the actual situation.

The information model, another view of system level, can be attained by collecting and organizing all the data, input and output contents especially, used by business activities or transformed between them in process model, which can be represented as table form or other structural forms. And the collected information should be divided into three types according to the structure of function view: intra-component information, which is used within a single functional unit/component and would be the design foundation of the component's computation; inter-component information, which shuttles across all the components in one sub-system and would be the blueprint for the design of the local application database using by this sub-system; and inter-sub-system information, which shuttles across the distributed sub-systems of the whole

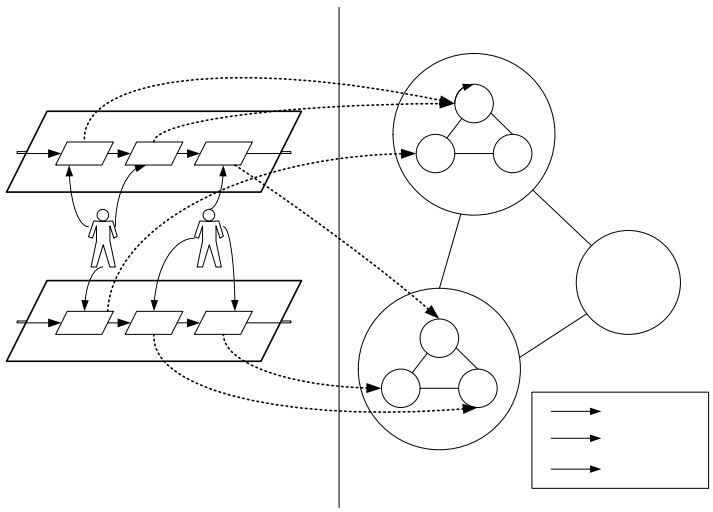

Fig. 3. the process and rules of mapping the business models to distributed system models

system and would be the blueprint for the design of the distributed database using by all the sub-systems.

The whole process of mapping the business models to distributed system models is summarized by Figure 3.

5 CONCLUSION AND FUTURE WORK

In this article, we propose an integrated enterprise modeling framework for developing enterprise-wide distributed system and a process of mapping the business model into the system architecture. Such process links the business with the system, and links the requirements with the design practically. On the other hand, the methods for unwrapping the enterprise viewpoint and the information viewpoint into another three viewpoints of RM-ODP and even into the program codes finally are relatively mature. So the supplement of this phase into the whole system development process would complete the lifecycle of the distributed system and assure the systems consistent with the enterprise, and with each other themselves.

In the future work, the formalization and automation of the extraction process are to be researched, and the solving mechanism for the conflict during the mapping is the dominant problem to guarantee the correctness and consistency of the developing process.

6 ACKNOWLEDGMENT

This work was supported in part by a grant from the National High-Tech. R&D Program for CIMS, China (Grant No. 2003AA4Z2020). [7]

REFERENCES

[1] ISO/IEC JTC1/SC21/WG7, “Open Distributed Processing—Reference Model—Part(1,2,3): (Overview, Foundations, Architecture),” International Standard ITU-T X.901 | ISO/IEC 10746-1, International Telecommunication Union (ITU) and ISO/IEC, Jun. 1995.

[2] OMG/ORMSC, “Model Driven Architecture,” OMG Document Ormsc, Jul. 1999

[3] F.B. Vernadat, *Enterprise Modeling and Integration: Principles and Applications*. New York: Chapman & Hall, 1996.

[4] Y. Fan, G. Wang, Z. Gao, *Introduction to Enterprise Modeling Theories and Methodologies*. Beijing: TUP, Springer, 2000. (in Chinese)

[5] F.B. Vernadat, “Enterprise Modeling,” *Production Planning & Control*, vol. 12, no. 2, pp. 107-109, 2001.

[6] Y. Zhao, Y. Fan, “Enterprise Modeling Framework for the Informatization Total Solution,” *Control and Theory*, accepted. (in Chinese)

[7] National High-Tech. R&D Program, “Research and development of the new generation ERP system”, technical report, 2003. (in Chinese)

[8] Y. Fan, “Enterprise development framework in the information age and total solution for information system implementation,” *The Aeronautical Manufacturing Technology*, no. 8, pp. 17-22, 2002. (in Chinese)

[9] Miller, T. H., Berger, D. W. *Totally Integrated Enterprises*. New York: St. Lucie Press, 2001.

[10] Y. Zhao, Y. Fan, “Research on Enterprise Modeling Based Implementation Method for the Total Solution of Informatization”, *The Aeronautical Manufacturing Technology*, accepted. (in Chinese)