Invited Paper

Simulation Modeling for Service-Oriented Development

Elena Orta[(1)], Mercedes Ruiz[(2)] and Miguel Toro[(3)]

(1) Department of Computer Languages and Systems. University of Cádiz (Spain)
E-mail: elena.orta@uca.es

(2) Department of Computer Languages and Systems. University of Cádiz (Spain)
E-mail: mercedes.ruiz@uca.es

(3) Department of Computer Languages and Systems. University of Seville (Spain)
E-mail: miguel.toro@lsi.us.es

ABSTRACT

As a consequence of the growing customer orientation of business, organizations are now showing a big interest in the development of technological solutions that allow them both to offer a quick respond to their customers and procuring operational efficiency. In this sense, SOA can be seen as a mechanism to help achieve the alignment of business goals and objectives with technology capability. SOA benefits an organization by abstracting business services from the specific technologies they are developed with. However, to translate SOA benefits into successful software development projects, well defined processes to develop software from business processes are needed. To achieve this, simulation modeling techniques can be useful tools since they make it possible to model and improve both business processes within organizations and their service-oriented development process. This paper presents a set of domains in service-oriented development where simulation modeling can be applied and discusses the benefits of this application. These domains are also related with the different phases of a general SOA lifecycle. An overview of the main works in this field is also included.

Keywords: Service-oriented development, simulation modeling, software developement process.

1- INTRODUCTION

Currently, business organizations are aware of the importance and benefits of aligning their business processes with the Information Technology (IT) that supports the automatization of their business processes. In order to make this alignment possible, it is required to

change the way business and the IT sector interact within an organization, so that, software applications can be developed from business process definitions allowing organizations to react agily to changes in the business domain. Nowadays, most business applications are still tighly coupled with the underlying IT infrastructure, in such a way that technological aspects such as data structures or programming languages restrictions, among other, still polute the definition of the business logic within computer applications.

Service Oriented Architecture (SOA) comes as a solution for this alignment problem. It provides the mechanisms to join the business domain with the IT one. In this sense, the software development process has been changing gradually in the last years to evolve from a rigid and centralized model to a decentralized, open and distributed one, centered on business processes. Business process orientation helps to achieve more agile and flexible architectures that can easily adapt to the continuous changes that occur in the markets where organizations make their business. The ultimate aim is to isolate the management of business processes from the applications that implement the logic of those business processes, so that changes in the business logic do not lead to changes in the application code. In this sense, SOA holds promise for business agility since it allows business process to change to meet new customer demands or market needs without causing a cascade effect of changes in the underlying IT systems. Thus, business agility comes hand in hand with IT flexibility.

However, to achieve the benefits of SOA it is also important to count with solid and defined processes to guide in all the phases of the management and development of SOA applications.

Simulation modeling techniques are currently considered as valuable tools that can be used for decision-making and software process improvement. In recent years, different contributions highlight the benefits that can be obtained from the application of these techniques [1]. In this paper, we propose an analysis of the different domains where simulation modeling techniques can be applied to service-oriented software development and the potential benefits that can be gained from this application.

The main reasons found to conduct this research are the following:

- The necessity to evolve towards methologies that make it possible to build software from business imperatives and business requirements instead of approaching this development under a strictly technological perspective.In this sense, SOA and its business oriented approach represent aconsiderable

advance.

- To study where and how simulation modelling techniques can be appliedin the field of service-oriented development and identify the potentialbenefits of this application.

- To study the applicability of simulation modeling techniques as complement tools for business process management systems in order to help in business process definition and improvement.

- The current small number of contributions found in the field of simulation modeling applied to the service oriented development.

This paper is organized as follows: Section 2 describes the main concepts regardingservice-oriented development and summarizes a proposal for service-oriented lifecycle methodology. Section 3 justifies the application of simulation modeling techniques to service-oriented development anddescribes the different domains where these techiniques can be applied as well as the potential benefits of this application. Section 4 provides an overview of the works developed in the field of simulation modeling and service-oriented development. Section 5 summarizes the paper and draws the conclusions and further work.

2-SERVICE-ORIENTED DEVELOPMENT

Marks and Bell [2] define SOA as “a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT nerwork. Services in SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages”.

One of the key factors when defining SOA is that it implies an architectural style independent of any technology. The concept of technology independence is not a new one in the field of software engineering. Structured programming in the 70s proposed the modular development approach as the first step towards the development of independent functions with well-defined interfaces. Structured programming acted as a forerunner for object-oriented programming that uses objects and their interactions to build software applications, bringing in new techniques such as inheritance, modularity, polymorphism and encapsulation. Next steps in software development are those provided by the component-based and distributed developments. One of the features shared by these approaches is that, at different stages of the software enginnering evolution, each one pursues the development of loose-coupling systems. In this context, SOA can be

seen as the further step in this evolution. Loose-coupling systems are considered useful where changes are frequent. Business agility is a crucial factor of success for current organizations. To provide a quick respond to changing requirements and customer requests is a real challenge for the IT sector, since it implies to develop an IT architecture ready to meet requirements that are unkown, as opposed to other previous development approaches. As a result, as companies move to adopt SOA, their application development process changes too.

2-1 LIFE CYCLE METHODOLGY

As stated before, service orientation is currently emerging as the new paradigm for developing distributed applications. In this new paradigm, services play a crucial role. According to [2], services are “reusable modular units of business capabilities, processes, or technical functions that are accessed and delivered in a repeteable fashion to consumers of those services”. Services, therefore, are the fundamentals elements for the development of application solutions. Under this approach, applications can be seen as sets of interactive and distributed services, and the IT capabilities of an organization as its body of services.

There are different forms of service-orientation architectures. The most popular and extended implements the concept of service as a web service. There are also other architectures that are gaining importance and popularity too. Among them, semantic web services, e-services and grid services are the prevalent techonologies.

Service-oriented applications are structured following a layered architecture where each element has a defined interface and a correspondence with one or more elements placed in the upper and/or lower level. However, although standarization of interfacing and protocols is a basic principle in the development of service-oriented applications, there is a lack of general agreement upon the definition of the lifecycle of an SOA application. Furthermore, even no mature methodologies for the analysis and design of these systems are available yet. As a consequence, different forms of service-oriented architectures have lead to different methodologies for their analysis, design and development.

Mostly of the current service-oriented development approaches come mainly from three sources:

- Existing object-oriented methodologies: Kunal Mittal [3] proposes the Service Oriented Unified Process (SOUP), which is a service-oriented development process that integrates elements coming from the Rational Unified Process (RUP) [4] and Extreme Programming[5].

- Component-Based Development: Papazoglou and Heuvel [6] propose a methodology for service-oriented software design and development based on RUP, Component Based Development (CBD) [7] and Business Process Modeling (BPM) [8].

- Agile methodologies: Zimmerman and colleagues [9] analyze how to apply the principles and techiques from Lean Software Development in service-oriented development. In [10] a review of different agile methodologies and their relation with service-oriented development is proposed.

There are also some contributions that offer proposals oriented for service analysis and design. The following are among them:

- Service Oriented Development of Applications (SODA). Defined by the Gratner Research Group, SODA proposes a collection of methods for service-oriented modeling, design and implementation [11].

- Method for Service Oriented Development (M4SOD). M4SOD proposes an approach to derive services from use case models at the business domain. It also determines how to implement the identified services. This approach is valid for composite as well as legacy applications [12].

For the purpose of this paper, we use the proposal for lifecycle methodology described in [13]. According to the authors, this methodology is "based on a roadmap that comprises one preparatory phase to plan development, and five distinct phases that concentrate on business processes". A brief description of the phases of this methology follows.

- Phase 1: Planning. The main objective of this phase is to assure the understanding of the business environment. It mainly consists of two activities oriented to detect the gaps between available and requested services, and the implications on cost, risks and benefits of developing new business processes.

- Phase 2: Analysis and Design. In this phase, services are identified and conceptualized using as an input the business requirements identified in the previous phase. After this, services are then designed and transformed into "a set of related, platform-agnostic interfaces".

- Phase 3: Construction and Testing. During the construction phase, services are implemented. This is a coding intensive phase. When the coding is over, service implementations need to be validated and verified in order to assure they meet the requirements and business process specifications

they come from.

- Phase 4: Provisioning. This phase collects a set of technical as well as business activities focussed on the support of service client activities.
- Phase 5: Deployment. During the development phase, the tasks associated with the publishing of interface and implementation details of the service take place.
- Phase 6: Execution and Monitoring. Finally, continuous evaluation of services performance and running business processes need to be monitored to assure the reactive mechanisms count with the necessary information to take adaptative actions.

3- SERVICE-ORIENTED DEVELOPMENT AND SIMULATION MODELING

3-1 BASIC CONCEPTS OF SIMULATION MODELING

A model is an abstraction or a simplified representation of a real or conceptual complex system. A model is built to show certain features of a system that need to be analyzed, estimated or controlled. As a consequence, a model includes some features of the system but not the whole system, only those features we are interested in.

A dynamic simulation model is a computational model that represents an abstraction or a simplified representation of a complex dynamic system. Simulation models offer as a main benefit the possibility of experimenting with different management decisions making it possible to analyze the effect of those decisions where the cost or risks of experimentation with the real system make it unfeasible.

The common objective of simulation models consists on providing the mechanisms to help predict the behavior of real systems and provide answers to the question “what if... ?”.

There are two broad types of dynamic simulation models: continuous and discrete-event. The distinction is mainly based on wheteher the state of variables in the model can change continuosusly or at discrete points in time. A continuous simulation model represents the interactions between key process factors as a set of differential equations that are solved by simulation rather than by analytical methods. Since our purpose is to model and visualize process mechanisms, continuous modelling will be the simulation modeling approach mainly used in this paper.

3-2 DOMAINS OF APPLICATION

In order to describe the different applications of simulation modeling in support of service-oriented development, we first categorize these applications into three basic nested domains: enterprise, project and application domains. The concept of working in domains in SOA development is not a new thing. Some companies as Oracle [14] also propose a definition of three scopes within which a SOA solution takes place. Figure 1 represents our proposal of nested domains for simulation modeling application.

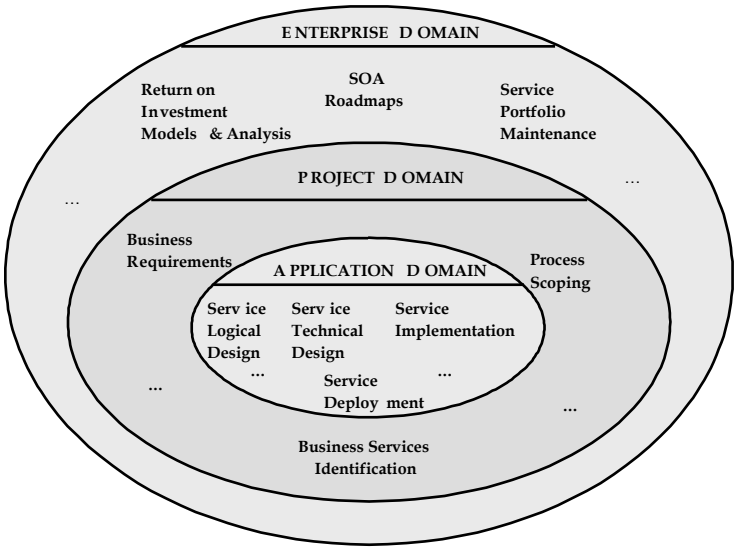

Figure1 Nested domains in SOA development

- Enterprise domain: This domain covers all the broad issues around SOA adoption within an organization. Issues such as return on investment or SOA roadmaps take place at this level. As it was mentioned before, we consider these domains as nested domains, being the enterprise domain the one that emcompasses the following ones. Simulation models can be used in this domain to study the overall service-oriented process. For instance, they allow organizations to study issues related to strategic planning and management, operational control, process improvement, technological change and all the aspects involving a deep understanding of the organization.

- Project domain: This domain covers all the issues related to the definition and development of the applications leading to the solution for the

business needs. This domain encompasses the application domain. Simulation models in this domain can help define and refine business and technical requirements, such as applications and services that can be necessary to satisfy the business requirements of the organization. In this sense, simulation models can be used together with Business Process Management Systems to simulate and, therefore, design improvement initatives for business processes that take place whithin organizations.

- Application domain: Finally, the application domain comprises all the activities that result in the development of single applications and/or services. In this domain, logical as well as physical solutions are developed. It is within this domain where technical design and implementation take place. Therefore, simulation models applied in this domain can help analyze different techonological issues such as optimizing service usage, the resources involved, service definition, security and exception handling, among others.

3-3 SIMULATION MODELING IN SOA LIFECYCLE

This section identifies the main activities in SOA lifecycle where simulation modeling can be applied as well as describes the main benefits that can be obtaind from this application. Table 1 summarizes the information presented in this section. For each domain and phase of the SOA lifecycle [13], we present where simulation can be applied and its potential benefits.

Domain / Lifecycle Phase	Enterprise	Project	Application
Planning	- Analyze Business Needs - Requirements Conceptualization - Finantial Analysis - Risk Analysis - ROI	- Project management	
Analysis & Design		- Process identification - Process scoping - Process realization analysis	- Service behavior specification
Construction & Test			-Dynamic testing of implementation
Provisioning		- Billing models selection - Service governance	
Deployment			
Execution & Monitoring		-Verification of Service Level Agreement conditions	

Table 1 Simulation modeling applied to SOA lifecycle and domains

3-3-1 ENTERPRISE DOMAIN

In this domain, we found simulation modeling can be useful at the following phase:

- Phase 1. Planning Phase.

According to [6], "the key requirement of this phase is to understand the business environment". Simulation models are useful in describing complex systems. Model building is considered a good exercise that leads to a better understanding of the system one is modeling. Therefore, simulation modeling can help in the analysis of business needs as well as in the requierements conceptualization, since a good understanding of the business needs is required in both activities.

Simulation modeling can be also of help in other activities of this phase that have a financial nature. For instance, [13] includes in this phase the activities regarding the analysis of business process realization scenarios. There are different strategies to do this according to [6], [13] and [15]. A simulation model can assist in decision-making as it provides help to analyze risks, Return on Investment or costs and benefits of different realization strategies.

Simulation modeling has been strongly used in the field of business models and practices. For instance, [16] presents the background and some of the lessons learned from a project in which system dynamics simulation was applied in the enterprise environment. Simulation modeling was used to examine business practices, validate corporate performance measures, train senior leaders in systems thinking, and to produce a forecast of long term profits and loss. [17] describes a simulation methodology, based on the combination of business objects and system dynamics that assists organizations in predicting future behaviors. The methodology eliminates the need for duplicating models for enterprise operation and simulation, and introduces a framework that enables the unification of the two in a single model. Further specific service-oriented applications in the field of service-oriented development are collected in Section 4.

3-3-2 PROJECT DOMAIN

In this domain, simulation modeling can be useful at the following phases:

- Phase 2. Analysis and Design Phase.

Being able to recognize functionality that is essential for the purposes of business processes is one of the key factors of the process identification activity of this phase. Once processes haven been identified, their scopes

need to be defined. Simulation modeling can help here in the definition of processes and their scopes as well as the relations of business process with external entities.

- Phase 4: Provisioning Phase.

The provisioning phase is a mixture of technical and business aspects for supporting service client activites. The main activites in this phase are: service governance, service certification and service metering and rating. Simulation modeling can be useful in service governance and service mettering and rating. Service governance collects a series of decision-making activities aimed at assuring whether the right types of services have been selected and whether all requirements for new services have been identified. Simulation modeling has demonstrated its usefulness in areas where decision-making is involved. This statement can also be applied to service rating since it is in this phase where billing models for services providers are negotiated.

- Phase 9. Monitoring Phase.

The key element of this phase is to obtain enough data about the running services in order to evaluate the meeting of service performance objectives. This set of metrics is also used to determine the level of cumpliment of the Service Level Agreement (SLA) with services providers. Also, this monitored progress serves as an input for the reactive mechanisms. Simulation modeling can be useful here to refine the definition for these reactive mechanisms.

As an example of the application of simulation in this phase, Tsesmetzis and colleagues [18] studies the problem of providers that receive concurrent requests of numerous customers for Web Services demonstrating different bandwidth and price properties. Although limited to two quality of sevice parameters (i.e. service bandwidth and cost), the paper makes a proposal “that aims to identify the services that should be delivered to the customers in order to maximize the provider’s profit, subject to maximum bandwidth constraints on the server side”. The authors use simulation to conduct experiments with their algorithm implemented in Mathlab, which is different to the type of simulation we are using in this paper. However, the results of [18] can be integrated in the equations of system dynamics models to be used in the monitoring phase of service-oriented development.

3-3-3 APPLICATION DOMAIN

- Phase 2: Analysis and Design Phase.

One of the key concerns of the design phase is to develop service specifications. According to [19], a service specification is composed of

three elements: structural specification, behavioral specification and policy specification. Simulation modeling can be of help here in better understanding services behavior as a complementary phase to behavior specification. It can also help to analyze the side effects of different service operations as a prior step to service behavior formalization.

- Phase 3: Construction and Testing Phase.

In the construction phase, the implementation of services takes place. This is a code development activity. However, once the services have been implemented the testing activities take place. In this group of testing activities, simulation modeling can be useful since it can help to test the implementation by simulating a real environment. Simulation can also help in testing of system performance, such as network congestions or system usability, for instance. Tsesmetzis's results [18] integrated with simulation models can be also used in this phase.

4- RELATED WORKS

This section presents some of the current contributions that apply simulation modeling techniques to model and simulate different aspects of service-oriented development.

– Enterprise Domain. Jeng and An [20] propose the use of dynamic simulation models in SOA project management. They present a framework for managing SOA projects and how system dynamics modeling simulation can enhance the effectiviness and agility of SOA project management. This work proposes a general lifecycle for SOA projects and includes two examples to illustrate the effectiviness of adopting this technique in managing large-scale SOA projects.

– Project Domain. In [21], the authors present dynamic simulation modeling as a complementary technique for business requirement identification. The main aim is to help and improve the tasks of business requirement identification. This improvement can be achieved by means of modeling and simulating the dynamic behavior of the business. [22] proposes a collection of heuristics and guidelines for the development of dynamic simulation models based on given business process models. The main idea consists on using the simulation models as tools to design and improve the effectiveness of business process models. The application of simulation modeling in the field of business process modeling makes it possible to study the features of this processes that are related with their stability as well as the identification of feedback loops within their control structures.

– Application Domain. [23] uses the simulation modeling techniques to present a framework for web service management. The aim of this framework is to optimize the use of wweb services and the resources involved. Simulation models can be placed at two different levels within web service management: the technical and the business level. Simulation modeling can be used in this context to show the interaction between service provider servers and priority rules collected in the Service Level Agreement (SLA).

5- CONCLUSIONS AND FUTURE WORK

In this paper, the application of simulation modeling techniques to the field of service-oriented development has been presented. After describing a general proposal for SOA lifecycle, a set of domains in service-oriented development where simulation modeling can be applied and its potential benefits have been described. A collection of different works developed in the field of simulation modeling and service-development has also been included. By using business process simulation, it is possible to better analyze process features such as resources involved, running times, costs, etc. in order to identify and improve process key factors.

Besides, simulation models can help solve questions regarding the integration of services and business processes, by helping in the identification and definition of service behavior.

Simulation models also make it possible to play simulation games that have no cost and allow managers to detect situations where, for instance, response time of a system can lead to an unsusccesful business.

Our future work is aimed at the development and validation of simulation models in the domains proposed in this paper.

ACKNOWLEDGMENT

This research was partly supported by the Spanish Ministry of Education and Science and the European FEDER funds under project TIN2007-67843-C06-04.

REFERENCES

[1] S.T. Acuña and M.I. S·nchez-Segura (eds.), New Trends in Software Process Modeling. Singapore: World Sicentific Publishing, Chapters 1-4, 2006.

[2] E.A. Marks and M. Bell, Service-Oriented Architecture: a Planning and Implementation Guide for Business and technology. Honoken, New Jersey: Wiley, 2006.

[3] K. Mittal, "Service Oriented Unified Process (SOUP)", IBM Journal, IBM, June 2005.

[4] P. Kruchten, Rational Unified Process – An Introduction, 3rd ed., Addison Wesley Professional, 2004.

[5] K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2nd. Edition, Addison-Wesley Professional, 2004.[6]M.P. Papazoglou and W.J. Heuvel, "Service-Oriented Design and Development Methodology", Int. J. of Web Engineering and Technology, Vol. 2, Issue 4, pp. 412-442, 2006.

[7] P. Herzum and O. Sims, Business Component Factory, Wiley, 2000.

[8] P. Harmon, "Second generation business process methodologies", Business Process Trends, Vol. 1, No. 5, May, 2003.

[9] O.Zimmermann,P. Krogdahl and C.Gee, "Elements of Service-Oriented Analysis and Design" http://www.ibm.com/developerworks/library/ws-soad1/

[10] P. Krogdahl, G. Luef and C. Steindl, "Service-Oriented Agility: Methods for successful Service-Oriented Architecture (SOA)development,Part1: Basics of SOA and agile methods" http://www128.ibm.com/ developerworks/webservices/library/

[11] S. Nigam, "Service Oriented Development of Applications (SODA) in Sybase Workspace", Sybase Inc. Whitepaper, Sybase Inc., 2005.

[12] Y. Kim and H. Yun, "An Approach to Modeling Service-Oriented Development Process", IEEE Internacional Conference on Services Computing, 2006.

[13] M.P. Papazoglou and W. Van de Heuvel, "Business Process Development Life Cycle Methodology", Communications of the ACM, Vol. 50, n∫ 10, pp. 79-85, October 2007.

[14] Oracle Corp. Oracle SOA Suite Developer's Guide 10g. Chapter 3, 2006.

[15] P. Brittenham,"Web-Services Development Concepts". IBM Software Groups, May, 2001. http://www.www-06.ibm com/software/solutions/webservices/

[16] J.F. Affeldt, “The Application of System Dynamics (SD) Simulation to Enterprise Management”, Simulation Conference Proceedings, Phoenix, Arizona, Vol. 2, pp. 1496-1500, 1999.

[17] A. Gregoriades and B. Karakostas, “Unifying business objects and system dynamics as a paradigm for developing decision support systems”, Decision Support Systems, Vol. 37, Issue 2, pp. 307-311, May 2004.

[18] D. Tsesmetzis, I. Roussaki and E. Sykas, “Modeling and Simulation of QoS-aware Web Service Selection for Provider Profit Maximization”, SIMULATION 2007, pp. 83-93, 2007.

[19] S.Johnston,“Modeling Se5rvice-Oriented Solutions” IBM DeveloperWorks, July,2005.http://www.128ibm.com/developerworks/rational/library/johnston/

[20] J.J. Jeng and L. An, “System Dynamics Modeling for SOA Management”. IEEE International Conference on Service-Oriented Computing and Application (SOCA’07), IEEE Computer Society, 2007.

[21] L. An, J.J. Jeng and C.E. Gerede, “On Exploiting System Dynamics Modeling to Identify Service Requirements”, IEEE International Conference on Services Computing (SCC'06), pp. 277-280, 2006.

[22] L. An and J.J. Jeng, “On Developing System Dynamics Model for Business Process Simulation”, Proceedings of the 2005 Winter Simulation Conference, pp. 2068-2077, 2005.

[23] L. An and J.J. Jeng, “Web Service Management Using System Dynamics”, IEEE International Conference on Web Services (ICWS'05), pp. 347-354, 2005.