Association for Information Systems
AIS Electronic Library (AISeL)

ICEB 2004 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-5-2004

A Component-based Framework for Distributed Business Simulations in E-Business Environments

Alexander Hampel

Edward W.N. Bernroider

Follow this and additional works at: https://aisel.aisnet.org/iceb2004

This material is brought to you by the International Conference on Electronic Business (ICEB) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICEB 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

A Component-based Framework for Distributed Business Simulations in E-Business Environments

Alexander HAMPEL [1], Edward W.N. BERNROIDER [2]

[1] Department of Computer Science and Business Informatics, University of Vienna, Austria
alexander.hampel@univie.ac.at
[2] Department of Information Business, Vienna University of Economics and B.A., Austria
edward.bernroider@wu-wien.ac.at

ABSTRACT

Simulations preserve the knowledge of complex dynamic systems and consequently transfer the knowledge of the cohesions of its elements to a specified target group. As the progress in information technology and therefore the dynamic e-business driven economy adapts even faster to the business demands, new ways to preserve this growing amount of knowledge have to be found. This paper presents an extensible business simulation framework which is realized as a component-based distributed Java Version 2 Enterprise Edition (J2EE) architecture. The framework aspires to offer an extensible and domain independent simulation environment which ensures the return of investment in the sense of implementing this framework once and extending it to the future requirements of diverse domains in e-business. The system architecture follows the requirements in offering distributed deployment of its components on highly standardized level by nevertheless staying vendor independent. The architecture itself was developed by model driven architecture (MDA)-conform software engineering methods using best of breed design patterns composed to a flexible micro-architecture which possess import facilities for simulation entities (business objects) and (business) processes from e-business solutions. Combining the features of the framework, the layered pattern driven micro-architecture, and the distributed J2EE architecture, the postulated knowledge transfer from rapid changes in e-business can be realized.

Keywords: business simulation, framework, architecture, J2EE, e-leaning, e-business

1. INTRODUCTION

Business simulations play a major topic in industry and higher education for more than 40 years. They preserve the knowledge of complex dynamic systems by modeling the cohesions and dependencies between the simulations' objects and transfer the modeled information by simulating its dynamic aspect on computer systems [1, 18]. In that way the knowledge about the behavior of the dynamic system is transferred to the specified target group [21]. Depending on what knowledge should be preserved, the cohesions of the system for the audience the simulation may differ. These advantages where identified by the American Management Association by developing its Top Management Decision Simulation in 1957. From 1957 till 1962, 89 different business simulations were developed by the industry, universities and other organizations [24]. In 1965 an accounting and book keeping simulation was developed to show the dynamic dependencies between the related records. Four types of simulation models were identified: strategic planning, internal planning, training program and research [19]. As a lack of computer systems these days, simulations were limited to non-executable implementations without having the ability to vary its modeled cohesions. As computer systems grew to solve more and more automated processes the awareness of linking these systems rose. The needs of integrating data and interconnecting networks (applications and systems) in business and industry was examined in depth to develop an open communication architecture [5, 29] and further more frameworks of for holistic information systems architectures were defined [33]. Middleware was and is the key player in integrating distributes systems starting 1994 with OMG's object-oriented open CORBA standard: a vendor, platform and program language independent specification [4, 15]. Integration approaches have widen the standardization process and brought new concepts of component ware , integration via web services [3, 7] and enterprise application integration (EAI) solutions for integration on the levels of data, application and processes [31, 23]. All the mentioned integration efforts have led to the new dynamic area of the e-business driven economy which is able to adapt even faster to the business demands [20]. As a matter of fast changes in e-Business life-cycle of virtual enterprises, new ways to preserve this growing amount of business and process knowledge, and transfer of the dynamic content have to be found. The following simulation framework is based on our very recent research project.

2. BUSINES SIMULATION FRAMEWORK

To set-up a business simulation without a framework on the meta-level would end up in a development without extensibility to adapt neither to new demands of the implemented business simulation nor the application of new simulation domains. These major disadvantages of having no continuous life-cycle for simulation systems led us to the development of a simulation framework to ensure a maximum of flexibility for the architecture applied as a corporate simulation and e-learning instrument. The following top-level requirements were defined for the framework:

- extensible to adapt to new demands
- enable distributed deployment
- simulation domain independence
- ability for import simulation objects, (business) processes and (business) rules from existing e-business or enterprise resource planning (ERP) systems

In asynchronous e-learning systems the reuse of implementation as extensible frameworks [6] is more common as in the area of synchronous e-learning solutions like business simulations. Models and implementations of business simulations are custom tailored to certain needs without developing a general simulation framework as [25] shows for the segment of production management. For our framework to comply with the stated requirements above is divided into the following dimensions of which the components of the simulation framework will be derived:

2.1 Simulation Client and Server Environment for Distributed Deployment

Monolithic architectures, as stand-alone database applications [18], bring along disadvantages which led to client-server architectures in distributed systems. As the simulation framework is deployed in distributed environment client and server components have to communicate via TCP/IP. All used components need to get access and be accessed by standard protocols using a multi-layered architecture. Further there must be the ability for a number of simulation systems to be interconnected.

2.2 Access with HTML-Browser and XML-based Client Applications

As distributed multi-layered simulation framework there has to be access to the server components in all expedient ways. HTML-browser access the simulation components with low featured graphical requirements like administration of the simulation system itself or registration for participants. Is a more featured graphical representation for providing a complex user interface for the simulation clients required, access with client applications and the use of XML communication has to be provided.

2.3 Domain and Domain Independent Components

To enable the framework to be simulation independent applicable there has to be a distinction between domain independent simulation facilities like communication services or operation figures and specific domain dependent components like the used simulation objects.

2.4 Production and Administration Environment for Separate Deployment and Assuring Knowledge Preservation in a Repository

Offering a domain independent open simulation framework, there has to be a central repository to save all modeled knowledge. As models of business simulations can be continuously extended while in use and simulations of different domains can be applied to the framework, the global repository plays an important role. This is an important feature for a corporate group of companies which wish to install the implementation of our simulation framework once and use an instance of the simulation framework for each subsidiary for different domains – preserving a holistic all-in-one mapping. International e-learning communities as university undergraduate schools could develop simulation models for different subjects and share the simulation models using the framework.

2.5 Synchronous and Asynchronous Communication Facility; Simulation Inherent and External Communication Facilities

Communication between the simulation parties and between interconnected simulation systems plays a crucial role in a distributed environment. Therefore synchronous communication facilities are needed for the simulation parties to chat or to send instructions between clients and the server infrastructure. Asynchronous communication is needed for all long-run activities where no immediate feedback is necessary which applies mostly for communication with external resources, e.g. feedback for the user registration process which is realized via SMTP.

2.6 Vendor Independence and Import Facilities from E-Business or EAI Solutions

To ensure a long life-cycle of the simulation framework vendor independence is an important issue. Some vendors use frequently update strategies of their products to keep the turn over high, to change their underlying architecture and APIs to stay in a proprietary way [9]. To adapt to new demands of the fast changing e-business, the simulation framework has to offer import facilities for simulation objects (business objects), (business) processes and rules, and from existing systems. As e-business solutions are set-up of different applications running on legacy systems, ERP systems or enterprise application integration (EAI) solutions, the import of these systems using standardized interfaces reduces the effort of creating new business simulations.

2.7 MDA-conform software engineering methods

As scientific long-term projects [18] and a large number of projects in the information technology have shown, the lack of using iterative and incremental development processes like specified in the MDA [22] or the Rational Unified Process (RUP) [26] leads to diverse negative side effects in multiple dimensions: (1) Lack of specification, (2) Lack of documentation, (3) Lack of conceptual and technical aspects as extensibility, performance and scalability. The key to develop sustainable, maintainable and extensible systems adequate engineering-methods. We use the RUP to transform requirements into implementation.

2.8 High-level Deployment of the Simulation Framework

The dimensions of the stated framework above are summarized as UML deployment diagram in a platform-independent model (PIM):

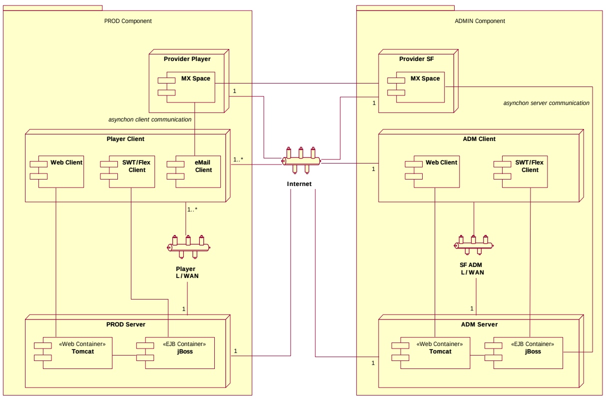

Figure 1: PIM deployment diagram of the simulation framework

3. PATTERN DRIVEN MICRO-ARCHITECTURE

System architecture is a complex task layering the architecture in different service subcomponents and in platform specific design issues. A micro-architecture is design pattern-based approach to set-up a holistic architecture to be extensible in the way of features and components, scalability and performance. It structures the patterns to interacting layers of subcomponents addressing services for local or remote objects [2]. Design patterns represent platform and programming language independent solutions for design problems in a certain design context and are a major development in system architecture [8].
The next step towards realizing the stated component-based simulation framework is to structure the system design into subsystems and service units to realize domain and domain independent components. As this is a very dynamic process it forces to use an iterative and incremental software engineering process like the model-driven architecture (MDA) [22, 32].
The server-sided domain independent simulation components in Figure 2 are as follows: *Accounting* (tracking any costs or transactions), *Communication* (between participants and client/server), *SimulationParty* (participants and simulation objects) and *System* (framework specific system tasks and process engine in the sense of specified workflows).

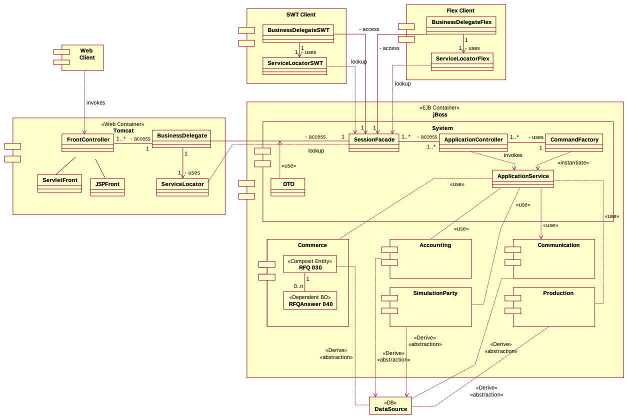

Figure 2: PIM component diagram of the micro-architecture of the core system (partial view)

The client-sided components communicate with the server via Simple Object Access Protocol (SOAP) [28] and offer a communication and simulation client framework. The simulation dependent components are *Production* and *Commerce* and were initialized by the domain of production control [17]. To structure the process of transition from the requirements of framework to the micro-architecture we order the relevant parts in the same sequence as we did in the section before:

3.1 Player Client and Server Environment for Distributed Deployment

In a distributed environment the processes of clients and servers have to be separately defined and assigned to the corresponding components. As there are no fat clients intended all major processes for verification of simulation rules are processed server-sided, only basic checking procedures to keep up with unintended mistakes during the simulation period are processed by the client. For this purpose a Session Façade [2, pp341] is used as façade [8, pp185] to decouple and limit client access to the framework and to reduce the network traffic using coarse-grained services. The facade incorporates with the Application Service [2, pp26, pp315] to fulfill use cases spanning business object access as described later on in the domain independent subcomponent.

3.2 Access with HTML-Browsers and XML-based Client Applications

To operate the simulation clients as remote clients a Business Delegate [2, pp302; 8, pp144] has to be implemented on the client side. As this component represents a remote object it has to provide a Service Locator [2, pp315] to lookup the server-sided delegate object. Each type of simulation client has to implement its own pair of business delegate and service locator. The communication between clients and server is done using different standardized protocols. HTML-browsers use HTTP/HTTPS, Java client applications use Java-RMI and SOAP/XML is implemented to extend the client type to Macromedia Flash applications [16] and other remote clients.

3.3 Domain Dependent and Domain Independent Components

Domain *dependent* components are business simulation specific objects which are realized as Business Objects [2, pp374] and designed as Composite Entity [2, pp391] as described in the J2EE [12] architecture below. Business objects are highly correlated with the import facilities of external systems and the simulation processes in this framework. As business objects unite business data and logic, they are on the one hand the key for mapping real-world or e-business objects to simulation objects in the way that they are the smallest entity, and no matter how the data is stored in the external system (e.g. split up into different tables), a composite view of the aggregated data is used as Composite Entity. The business logic stored as business processes on external systems and can be imported too. The business objects are aggregated to subsystem components and encapsulated on a higher abstraction layer. The import with direct access to the external systems (e-business, EAI solutions or ERP systems) using J2EE specifications is described later on. Export facilities are not provided in this framework nor the use of Learning Object Metadata [11] for the internal persistence layer as these features are no central issue in our research project.
Domain *independent* components are independent of the business simulation like the system boot-up process, loading, creating, importing and editing of simulations or the administration of the simulation environment. Further the communication objects and the accounting including operating figures to evaluate simulations are domain *independent*. They are on the other hand the key to the dynamics of business simulations: processes belonging to the business object are encapsulated as object's method and processes spanning various business objects are aggregated to use cases and combined as Application Service which represents the domain *independent* simulation engine.

The access to the persistence layer is managed by the J2EE application server [12] and non-DB sources by Data Access Objects [2, pp462]. Data Access Objects abstract and encapsulate the access to the persistence layer in the way that business objects do not have to have information about the persistence store and can transparently access all kinds of data like XML feeds or flat files.

3.4 Production and Administration Environment for Separate Deployment and Assuring Knowledge Preservation in a Repository

The framework is designed to be used as multi-deployment implementation with a distinction between the production systems and the administration system containing the global repository (Figure 1). As there is the need of a central repository to back-up the simulation objects, the business processes and to preserve the modeled dependencies of all production systems, it is further used for documentation purposes. The set-up of new simulation models and the transfer to other simulation instances of this simulation framework is managed by the administration environment.

3.5 Synchronous and Asynchronous Communication Facility; Simulation Inherent and External Communication Facilities

To enable simulation inherent client to client, client to server communication, and external communication facilities, the communication component is part of the domain *independent* simulation components. As there is a multi-view to the components of the business simulation framework and the micro-architecture, these views overlap. Synchronous and asynchronous system communication is realized with the standard platform API. To summarize at this point for the J2EE architecture: Java-RMI is used for object calls, SOAP/XML for client-server and server-server communication and SMTP for e-mail.

3.6 Vendor Independence and Import Facilities from E-Business or EAI Solutions

Vendor independent development is realized by using a J2EE architecture as it is vendor, operating system and hardware independent. Depending on the application demands the simulation framework can be deployed on small, medium or large scale systems, standalone or clustered. The distributed J2EE architecture offers a connector architecture (JCA) [13] to enable all major platforms and vendors to provide their own resource adapters or implement the needed JCA connector with minimal interface programming themselves. Further as stated in the requirements for the framework, the only way to deliver an extensible system to adapt to new demands is to lengthen the life-cycle of the application by using sustainable and open standards. Our simulation framework therefore offers import facilities using JCA or indirect import of any desired semantics using XML or CSV format.

4. DISTRIBUTED J2EE ARCHITECTURE AND ITS IMPLEMENTATION

The sections above described the requirements, the deployment and the micro-architecture (both PIMs) of the framework. Following the MDA the next step towards the implementation is the transfer of the PIM to a Platform Specific Model (PSM) [22] to ensure the system architecture keeps the requirements: extensible, distributed deployment, simulation domain independent and import capabilities. The chosen J2EE [12] platform and its implementation specific topics follow.

4.1 Basic Rules for Transformation of the PIM to PSM J2EE Architecture

To ensure an adequate and coherent transformation from the PIM to the PSM some basic rules were set up:

1. Application of all the capabilities of the J2EE server: no server-sided plain old java objects (POJOs),
   no bean-managed persistence (BMP)
2. Favour declarative over programmatic solutions: use the mightiness of the deployment descriptors
3. Use sustainable and open standards for all interfaces: SOAP/XML for client-server and server-server communication

4.2 Design of the Persistent Domain Model

Following the basic transformation rules the data access to the persistence layer is implemented using container-managed persistence (CMP) thus the container requests, fetches and manages the life-cycle of the entity beans. This reveals performance advantages, bean pooling optimization, and no mix up of framework code and code for the persistence layer.

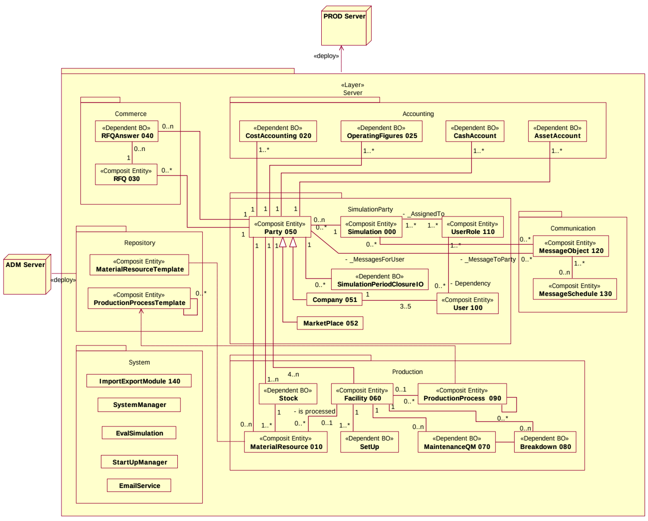

Figure 3: PSM of persistent objects as J2EE design model for implementation (partial view)

The relationship between persistent objects are defined declarative in the deployment descriptor and are managed by the J2EE container: referential integrity, cardinality, cascading delete, and using EJB Query Language (EJB QL) for portable query code [12]. To ensure performance, extensible and scalability, the components and its business (domain) objects are implemented as Composite Entity [2, pp 391] which enables the object-oriented aspect of capsulation of business objects further more. For non-database persistence like XML feeds, legacy systems, LDAP repositories or flat-files, Data Access Objects [2, pp 460] are implemented.

4.3 Import Facilities using JCA and JMS

Using the J2EE technology platform enables synchronous and asynchronous access via JCA and Java Message Service (JMS) [14] for the import of data and process information of existing e-business or ERP systems. The JMS API enables direct access to external JMS-compliant systems and the implementation of the JMS import facilities manifest as minor expenditure. JCA resource adapters are available for all major systems as middleware infrastructure like IBM MQSeries [10] or ERP systems like SAP's new Netweaver Architecture [30] and can be plugged-in for data import as needed.

6. CONCLUSION

The presented component-based framework for distributed business simulations in e-business environment offers a new conceptual way as an extensible micro-architecture to implement this framework once and extending it to future requirements of diverse domains for simulation and e-business solutions. The pattern driven micro-architecture ensures further development of the framework without pitfalls of scalability or performance problems. We believe new e-learning solutions for preserving and transferring the inherent dynamic system knowledge of rapid changes in e-business will adopt similar strategies as the frameworks introduced in this paper. Future research in combining our simulation framework with Learning Object Metadata for export to other simulation and e-learning systems or for direct access for the persistence layer would widen the framework for even more interoperability.

REFERENCES

[1] Albores- Barajas, A. et al., "Generic Business Process Models for E-Business in Manufacturing companies: is simulation useful?", *Engineering Management Conference*, IEEE, 8-20 Aug. 2002, Vol.2, pp685-690, 2002.

[2] Alur, D. et al., "Core J2EE patterns – Best Precice and Design Strategies", Upper Saddle River: Sun Microsystems/Press, Pentrice Hall, 2003.

[3] Arsanjani, A., "Developing and Integrating Enterprise Components and Service", *Communications of ACM*, Vol. 45, No. 10, pp31-40, 2002.

[4] CORBA Home, www.corba.org

[5] Cypser, R.J., "Evolution of an open architecture", *IBM Systems Journal*, Vol. 31, No. 2, pp161-188, 1992.

[6] Derntl, M., Mangler, J., "Web Services for Blended Learning Patterns", *Proceedings of 4th IEEE International Conference on Advanced learning Technologies (ICALT'04)*, August 30-Sept 1, Joensuu, Finland, 2004.

[7] Fremantle, P, et al., "Enterprise Services", *Communications of ACM*, Vol. 45, No. 10, pp77-82, 2002.

[8] Gamma, E. et al., "Design Patterns – Elements of Reusable Object-Oriented Software", Indianapolis: Addison-Wesley, 1995

[9] Hampel, A., Bernroider, E.W.N, "Strategic Software Engineering Decisions Concerning the System's Architecture and the Development Environment for an Internet-Based Business Simulation", World Conference on Educational Multimedia, Hypermedia and Telecommunications, Vol. 2004, No. 1, pp. 64-65, 2004.

[10] IBM, "Patterns for e-business", http://www-106.ibm.com/developerworks/patterns/u2b/at2-product-w2k.html

[11] IEEE , "Draft Standard for Learning Object Metadata", IEEE 1484.12.1-2002 (LOM), 2001.

[12] J2EE, Java Version 2 Enterprise Editions, http://java.sun.com/j2ee

[13] JCA, Java Connection Architecture, http://java.sun.com/j2ee/connector

[14] JMS, Java Message Service, http://java.sun.com/products/jms

[15] Lewis, G.R., "CORBA 2.0: universal networked objects", ACM StandardView, Volume 3, Issue 3, pp102-106, 1995

[16] Macromedia, Flash MX, http://www.macromedia.com/software/flash

[17] Matyas, K., "Planspiel Produktionssteuerung - Simulation Game in Production Control", PhD Thesis, Austria , Vienna University of Technology, 1989.

[18] Matyas, K., Hampel, A., "Internet-based Business Simulation for the International Education of Industrial and Systems Engineers – An Interdisciplinary Scientific Long-Term Project", *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Vol. 2003, No. 1, pp. 872-873, 2003.

[19] Mead, G.C., "Business Simulation – Dynamic Tool or Enjoyable Pastime?", *Journal of Accountant*, No. 121, pp84-85, 1966.

[20] Nayak, N. et al., "Virtual enterprises: building blocks for dynamic e-business", Australian Computer Science Communications, Vol. 23 No. 6, pp80-86.

[21] Nurmi, S., Lainema, T, "Facilitating the Unterstanding of Business Processes with Simulation Game", World Conference on Educational Multimedia, Hypermedia and Telecommunications, Vol. 2003, Issue. 1, pp. 2163-2170, 2003.

[22] OMG, Model Driven Architecture, http://www.omg.org/mda

[23] Ortiz, A. et al., "Enterprise Integration – Business Process Integrated Management: a proposal for a methodology to develop Enterprise Integration Programs", Computers in Industry, 40(1999), pp155-171, 1999.

[24] Patten, Th.H., "Business Simulation in Industry and University Education", *Personnel Journal*, No. 41, pp566-567, 1962.

[25] Riis, O. "Games for Production Management", http://iprod.auc.dk/x-proj/gamespm/www-games. html, *International Workshop of the IFIP WG 5.7: Experimental Interactive Learning in Industrial Management.*, 22-24 May 2003, Aalborg, Denmark, 2003.

[26] RUP, Rational Unified Process, http://www. rational.com

[27] Singh, N., "Unifying Heterogeneous Information Models", Communications of ACM, Vol. 41, No.5, pp37-44, 1998.

[28] SOAP, W3C XML Protocol Working Group, http://www.w3.org/TR/soap

[29] Stecher, P., "Building business and application systems with retail application architecture", *IBM Systems Journal*, Vol. 32. No. 2, pp278-306, 1993.

[30] Stiefbold, O,. Platzer, U., "What's New in SAP's Technologies?", Javamagazin, 6/2003, pp45-51, 2004.

[31] Sutherland, J., Heuvel, J., "Enterprise Application Integration and complex adaptive systems", *Communications of ACM*, Vol. 45, No. 10, pp59-64, 2002.

[32] Wang, H., Zhang, D, "MDA-based Development of E-Learning System", *Computer Software and Applications Conference*, COMPSAC 2003. Proceedings. 27th Annual International , 3-6 Nov. 2003 , pp684-689, 2003.

[33] Zachmann, J.A., Sowa, J.F., "Extending an formalizing the framework for information systems architecture", *IBM Systems Journal*, Vol. 31. No. 3, pp590-650, 1992.