The MODIFY Project: Combined Business and System Modeling for Adaptable Enterprise Computing System Design

(Position Paper)

Artur Caetano António Rito Silva José Tribolet João Neves Pedro Sinogas

{artur.caetano, rito.silva, jmt}@inesc.pt {jneves, psinogas}@ceo.inesc.pt

INESC - Instituto de Engenharia de Sistemas e Computadores
Organizational Engineering Center and the Software Engineering Group
Rua Alves Redol, 9
1000-029 Lisboa, Portugal

Abstract

This paper presents the research perspective of INESC's Organizational Engineering Center and the Software Engineering Group regarding the modeling and design of adaptable enterprise computing systems that support dynamic business requirements. We describe the objectives and working methodology of the MODIFY project, which aims at defining a component-based solution to the design of enterprise computing systems using a combined business and system model.

Keywords: business engineering; business modeling; software engineering; adaptability; component composition.

1. Introduction

Large-scale business competition and a shift to customized products and shorter life cycles have created a setting of continuous business structure change. To effectively compete, businesses are constantly adapting and restructuring its structural elements, such as processes, products and suppliers. A successful information technology strategy must enable business flexibility by providing solutions at a rate which offsets for the increasing pace of business structural change.

However, a solution that only defines plans and control activities to reach the business goals is prone to fall short of its objectives since the means to accomplish the goals and the goals themselves change over time. Moreover, a coherent strategy will only represent a noteworthy competitive lead if it is not built on standard business practices and processes. This requires an organization to be able to learn and adapt faster and in a more efficient manner than its competitors. One way to attain such adaptability and to manage the complexity of handling the organization's processes is to exploit the capabilities of enterprise information technology. Despite a significant amount of approaches related to business and software engineering which try to explore enterprise information technology, these answers have repeatedly failed to accomplish the expected gains on cycle time reduction and overall efficiency increase so that change is supported, meeting the evolution of the business requirements.

On the business engineering area, for example, the existing business modeling methodologies tend to disregard the intrinsic dynamics of the organization and the resulting business models are not easily adapted to the system domain requirements, thus not supporting an effective implementation of the system. On the software engineering area, a similar problem exists since there are no models or design methodologies that explicitly provide a connection back to the business domain, thus making difficult to identify and support the business requirements and to design flexible software solutions. As a result, in an environment where the business requirements and processes are continuously adjusting themselves to the business needs, the supporting software systems either become obsolete or gradually hinder the business structural improvements, as the changes in the system become more time and effort consuming.

As business complexity increases, there is a negative impact on correct requirements interpretation, solution reliability, overall business flexibility and cost. Therefore, the challenge to both business and software engineering is to provide adaptable and rapid solution deployment of accurately specified business requirements.

2. Research objectives

The current research objectives of the joint collaboration of INESC's Organizational Engineering Center and the Software Engineering Group concerns the modeling and design of adaptable enterprise computing systems that effectively support dynamic business requirements. This issue raises a large amount of research questions in the areas of business and software engineering. To provide answers for this complex problem, we have initiated the MODIFY project that focus on the following issues:

- Defining an appropriate yet simple characterization of business concepts and objects for business domain modeling.
- Describing a model-based specification of the business and system domains.
- Using component-oriented concepts in the design of enterprise computing systems.
- Exploring component composition and configuration.

An adaptable software system can be adjusted to changing and multiple requirements. In the context of enterprise computing, the successful implementation of the underlying software systems directly depends on the system to be related with the business processes it supports and in the ability to reconfigure the system in the dynamic organizational environment. The key property of our approach is to provide a clear mapping between the business and system domain components while preserving the compositional structure of the system. This requires the model to be two-way traceable, meaning that it is not only connected back to the business domain, which are the basis of the system requirements, but also associated forward to the system domain and the corresponding software architecture. Additionally, this model makes possible the inspection of the business and system components so that the analysis and optimization of both business and system domains becomes possible.

In the next sub-sections we describe the guiding principles of the MODIFY methodology, depicted in Figure 1.

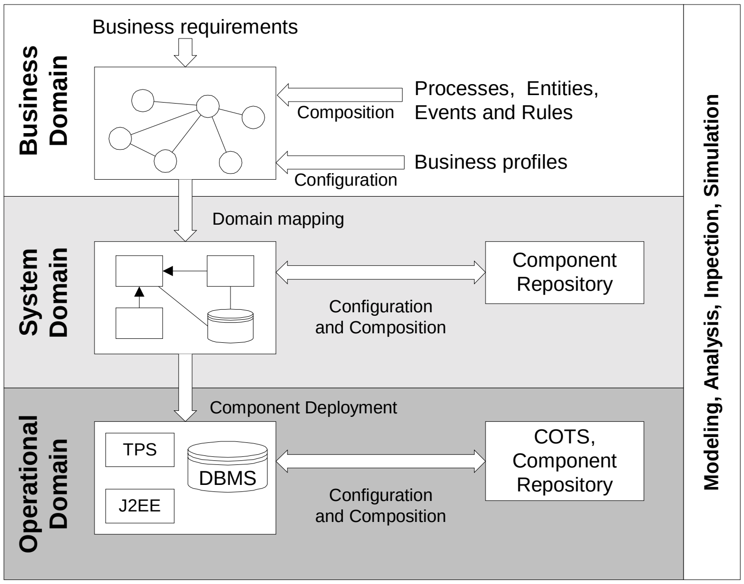

Figure 1 The MODIFY methodology.

2.1. Business domain modeling

The ability to describe the business abstractions in the business model is crucial, on the one hand, for achieving adaptability to business process changes and traceability to the system domain, and on the other, to communicate the business model to IT and management staff. Besides defining and representing the business abstractions, the business model should also contribute to facilitating knowledge sharing, simulating "what-if" scenarios, and comparing the business structure with "best practices" [Malone 99].

The definition of business abstractions can exploit domain ontologies and business patterns [Kilov 94, 99, Eriksson 00, Marshall 00, Sims 00]. However, to keep the model simple but extensible, it is necessary to identify a set of concepts from where complex or more specific business conceptualizations can be derived. We believe these concepts to be business entities, processes, events and rules [CBO 97, BOI 98]. A *business entity* is a representation of an element of significance to the business, operated during business process execution. It can be a person, place, thing or concept relevant to a business. A *business process* is the definition of a cohesive behavior of an enterprise in its interaction with the business entities. A business process can be defined in terms of steps controlled by business rules that are triggered by and can generate business events. A *business event* is an occurrence of interest which is relevant to the execution of the business processes. Finally, a *business rule* is a statement that defines or constrains some feature of the business.

Special attention is required when defining the concept of business entity since it usually characterized as an otherwise immutable element, except for having a lifecycle expressed as set of discrete states. A richer entity definition must explicitly support the change and evolution produced by time. This is particularly important when modeling and optimizing the usage of human resources, since the evolution of roles turns out the description of these resources to be a complex task [Gibson 99].

2.2. Model-based specification

Using a model-based specification of the business and system domains allows for a clear separation of concerns between the business and the implementation neutral system concepts. This specification expresses the semantics of business and system domain elements and how intra and inter-domain relationships are defined.

The model must assure that the semantic relationships existing at the business requirements level are kept after the transformation to the system domain. This requires the definition of conformance rules which describe the semantic mapping at each abstraction layer. In addition, it shall be possible to inspect and analyze the deployed components and identify the corresponding business rules. The opposite procedure shall also be possible, i.e., identifying the components and corresponding relationships that implement the elements of the business structure. As stated before, this requires the components to be traceable from the business to the system domain and vice-versa.

Components, as encapsulated units of functionality, interoperability, reuse, and composition, are attractive for adaptable software design since they allow application-dependent concerns to be disassociated from adaptability concerns. Moreover, the ability to compose and configure business and system components not only simplifies solution development and promote reusing but also allows several layers of functionality and complexity to be defined. This allows the underlying complexities (e.g. software technology issues) to be concealed from developers who are engaged in the construction of business-oriented solutions.

The model will also define component composition rules that specify how adaptability and reuse can be achieved within the model specification. A set of corresponding design patterns shall favor component reuse, configuration and dynamic reconfiguration.

3. Project timeline

We are expecting to present the first research results later this year in relation to a business ontology for business domain modeling.

For more information, visit the MODIFY web site at http://ceo.inesc.pt/modify/

4. Acknowledgments

The MODIFY project is partially funded by grant FCT-MCT 34565/99.

5. References

[BOI 98] "Roadmap for the Business Object Initiative: Supporting Enterprise Distributed Object Computing". OMG Document bom/98-12-04.

[CBO 97] Common Business Objects Version 1.5, BODTF White Paper. OMG Document bom/97-12-04.

[Erisksson 00] H.-E. Eriksson, , M. Penker. "Business Modeling with UML – Business Patterns at Work". Wiley, December 1999

[Gibson 99] N. Gibson, C. Holland, B. Light. "Enterprise Resource Planning: A Business Approach to Systems Development". 32nd Hawaii International Conference on System Sciences, 1999.

[Kilov 94] H. Kilov, J Ross. "Information Modeling: An Object-Oriented Approach". Prentice-Hall, 1994.

[Kilov 99] H. Kilov. "Business Specifications: The Key to Successful Software Engineering". Prentice-Hall, 1999.

[Malone 99] T. Malone, K. Crowston, J. Lee, B. Pentland *et al.* "Tools for inventing organizations: Towards a handbook of organizational processes". Management Science, March 1999.

[Marshall 00] C. Marshall. "Enterprise Modeling with UML: Designing Successful Software Through Business Analysis". Addison-Wesley, 2000.

[RM-ODP] Reference Model of Open Distributed Processing (RM-ODP). ISO/IEC 10746.

[Sims 00] P. Herzum, O. Sims. "Business Component Factory – A Comprehensive Overview of Component-Based Development for the Enterprise". Wiley, January 2000