

AN ONTOLOGY-BASED APPROACH FOR COMPLEX SOFTWARE SYSTEMS EVOLUTION PLANNING

On the example of B2B systems

Yulia Madorskaya

St.-Petersburg State Polytechnic University, Department of Central Research
Institute of Robotics and Technical Cybernetics

Polytechnicheskaya Street, 29
195251 St.-Petersburg, Russia
e-mail address: mail@satur.ru

Abstract

B2B systems are a striking example of a complex multicomponent system, evolutionary maintenance of which the most dramatically illustrates the problems of coordinated changes. The foundation for coordinated modifications is reliable planning. This paper describes the key problems of evolution planning of complex software systems and proposes a comprehensive ontology-based conception which is addressed to the entire set of the problems. This conception defines design principles and requirements to all components of the solution including metaontology for the traceability model. The solution satisfying these requirements have been developed, tested in projects including B2B system development and presented in this paper.

INTRODUCTION

B2B systems are automated systems, designed for integration of industrial clusters. They are important tools for enhancing the competitiveness

of each enterprise in the cluster and the cluster as a whole.

Desired improvements can be achieved in the case that B2B system evolves in time according to the needs of enterprises in the cluster. Simultaneously, the enterprises in the cluster must rapidly adapt their processes to comply with decisions taken at the level of the integrating system. *Coordinated* modifications of the business processes and corresponding B2B functions are crucial to the viability of the cluster.

The foundation for coordinated modifications is reliable planning. At the same time, in the conditions of growth of software systems complexity, planning is carried out with significant inaccuracy. This is well known and clearly confirmed by the Standish CHAOS reports (for example [3]), which show that in the planned terms holds less than 40% of all projects. To understand what specifically prevents the reliable planning it is necessary to carry out in-depth study of modifications planning process and to see what lies behind the definition of effort and duration during the plan preparation.

ANALYSIS OF THE MODIFICATIONS PLANNING PROCECSS

The key task, which accuracy considerably limits reliability of modifications planning, is the estimation of change propagation and their complexity. This estimation is the key information for the planning process. It is used to calculate work and duration for modifications, to decide whether to implement a change request or not, to choose between different implementation ways.

Moreover, such estimation is conducted not only for the original change request but iteratively determined for all derivative design decisions, which are essentially the same change requests. Thus, even a slight inaccuracy of the estimation can lead to significant deviations in the plan, since errors are being accumulated.

The estimation of change propagation and their complexity for B2B systems is based on the analysis of system models representing requirements. During this analysis design decisions are undertaken in order to determine the direction of change. So as it's based on the design process the estimation of change propagation is a weakly formalizable problem.

During the planning process the estimation is always held on incomplete data as in terms acceptable to decision-making it is not possible neither to undertake a full design cycle nor collect all data necessary for the analysis. Therefore accuracy and time of the estimation strongly depends on the quality of the data and experience of an expert who through the incomplete data should assess the whole picture of changes.

THE KEY PROBLEMS

Analysis of development practices shows that there are four groups of key problems which affect accuracy of the estimation and respectively affect reliability of planning. They are related to requirements analysis during the estimation.

Greate volume, diversified and multi-dimensional representation of the data at every stage of the life cycle.

The modular approach is used while developing such complex system as B2B and work is distributed between different teams. Team's choice about system description methods depends on the stage of life cycle, peculiarities of the domain and especially depends on their habits, experience and qualification. In accordance with the software design technology adopted in a specific team, the system requirements can be presented in the form of descriptions of various types (plain text, UML, IDEF, ARIS, etc.). So the real overall design model of a complex software system is the composite description, which is created by using different languages and a vast number of heterogeneous meta-concepts. To accurately estimate change propagation through the whole model is a challenging and time-consuming task.

Poor accessibility of the data to the joint analysis.

- The data are distributed over the different repositories.
- The descriptions cover the system not completely.
- The relationships between elements of the descriptions are not clearly specified.
- The descriptions cannot be unambiguously understood in the absence of the developer.

Incomparability of the data.

Different teams use and adopt various design methods so descriptions produced according to each particular technology are not easy matching and comparing while this is essential for the overall analysis.

Errors in the data and relations between them.

It is well known that there are many errors in the system descriptions, including outdated elements and relationships, duplicates, inconsistencies, and so on.

Enhancing each other all mentioned problems ultimately lead to a

significant reduction in the accuracy of the estimation.

In-depth analysis of requirements traceability methods which is the base for the propagation analysis has discovered a number of lacks from the point of view of the practical use in the planning process:

- Dependence on design paradigm and development technology.
- Ambiguous semantics of the categories and relationships.
- Excessive for the planning process the number of categories and relationships.
- Lack of means for preventing errors.

Other problems are outlined in [1], [2].

The most of existing methods are aimed primarily at identifying traceability relationships between categories of a specific design technology, such as UML. They do not provide the basis for comparing and tracing through all data produced by different teams with different technologies while this is essential for planning.

Only the comprehensive solution of problems A-H can increase reliability of planning and improve the coordination of modifications for such complex systems as B2B.

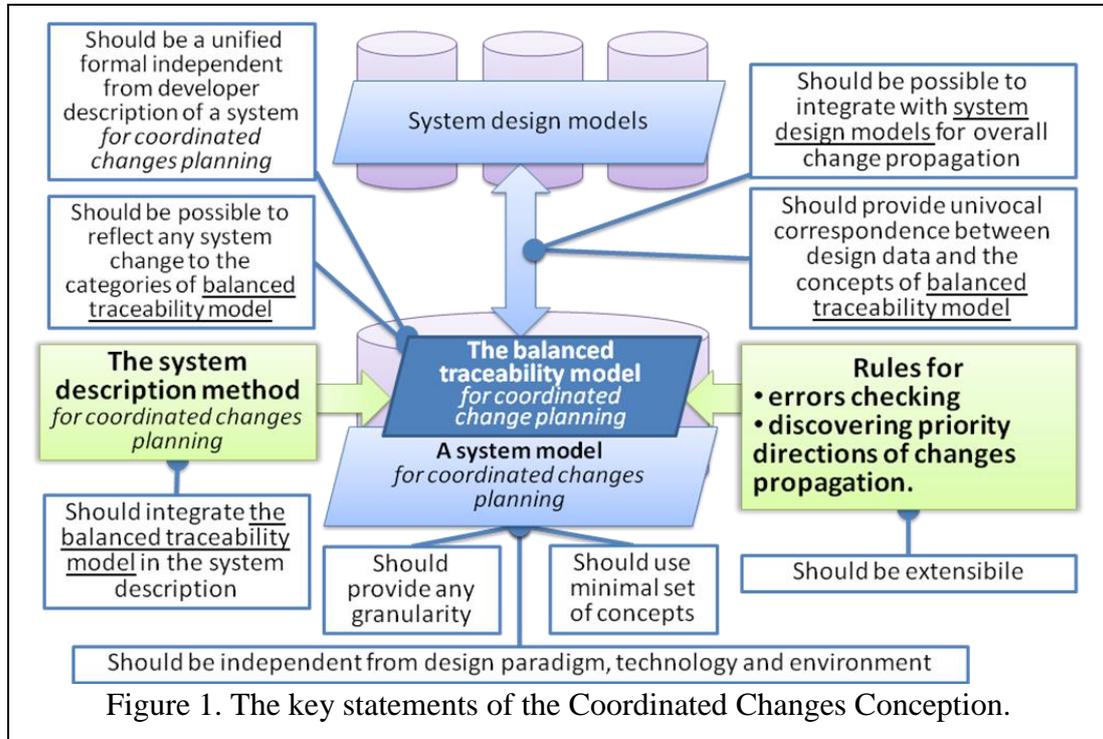
A COMPREHENSIVE CONCEPTION FOR CONSTRUCTING A CHANGE ESTIMATION METHOD FOR MODIFICATIONS PLANNING

It's widely accepted that ontologies can be used as a knowledge management mechanism to represent multi-perspective software artifacts in a common way for interoperability and traceability purposes [5]. We state that requirements management in general and requirements traceability in particular can significantly benefit even more from the use of other components of formal ontologies such as axioms and inference rules. They could be effectively used to prevent requirements errors and to automatically discover priority traceability directions. This work addresses the traceability method which uses all components of formal ontology to improve change estimation.

We propose a comprehensive conception which takes into account all four early mentioned groups of problems. This conception defines design principles and requirements the solution components which aim is to improve change estimation while focusing on the planning process. The key points of the conception are presented in Fig. 1 and briefly described below.

The basic component of the solution should be the model of a software ontology which in fact is a traceability model. This model should be *balanced* for planning process. The categories and relationships which this model defines should be used to construct a specific ontology of a system – *a system*

model for coordinated changes which should be used to estimate change propagation. That it's why we call it "model of ontology" in contrast to "ontology".



To be balanced for the planning process such an ontology model must meet the following requirements. It should include categories and relationships to describe a system at the upper-level - to provide comparability of the estimation through the entire software life cycle and through specific design technologies of different teams this ontology model should include the minimal set of concepts which are independent of design paradigm, technology and environment. Minimizing the number of categories, which are essentially the primary units of estimation, will provide for an expert the possibility to get more experienced in translating the such estimation into units of work and duration. The chosen set of categories and relationship should provide one-to-one correspondence of design artifacts to these entities, be sufficient to reflect any changes in the system and define rules for error checking.

Thence *the system description method* should be developed to provide construction and maintenance of *the system model* according to *the balanced traceability model* satisfying above requirements. Such a method should integrate *the balanced traceability model* in a system description.

The comprehensive solution should include the following formalized and integrated components: *a system description method, rules for errors*

checking and rules for discovering priority directions of changes propagation. Where categories and relationships of *the balanced traceability model* should be used to develop *rules (axioms) for errors checking and inference rules to discover priority directions of changes propagations.*

All components of the solution should be extensible. This may improve the reliability of planning by taking into account rules of certain domain.

Joint implementation of above requirements will raise accuracy of the estimation of change propagation that in turn will improve the reliability of planning.

Based on the proposed conception the solution fully satisfying the specified requirements have been developed. This solution contains the following interrelated components:

- Harmonized definition system which formally defines *requirement* through the logic calculus terms, covers any requirements classes and establishes requirement key properties necessary for design errors reduction [6].
- Formal classification of requirements errors which are essential to define rules for errors checking [7].
- The balanced traceability model which consists of two parts – traceability triangle – software traceability model independent of software class and its extension for Enterprise system development. These models define the structure of system descriptions for coordinated changes [7].
- The changes estimation method for coordinated changes based on the previous components and consisting of the system description method, rules for errors checking and inference rules for discovering priority directions of changes propagation united in formal system [8].

In this paper we present the part concerning the traceability model.

TRACEABILITY TRIANGLE AND ITS EXTENSION FOR ENTERPRISE SYSTEM DEVELOPMENT

To construct *the balanced traceability model* which will provide one-to-one correspondence of its categories with design data we've studied a large number of regulations which define requirements structures in different ways. There were architecture frameworks, design methods, development methodologies, best practices and ontologies. We defined generalizations for their categories, found their intersections and at last united the resulting intersections. In such a way we found the minimal layer of abstraction which satisfies the above requirements. Than we add appropriate relationships which are necessary to discover change propagations and to manage granularity of descriptions.

The result of this investigation is the traceability triangle (TT),

(presented at the Fig.3), an ontology model which defines minimal set of concepts and relationships satisfying the above requirements. It consists of

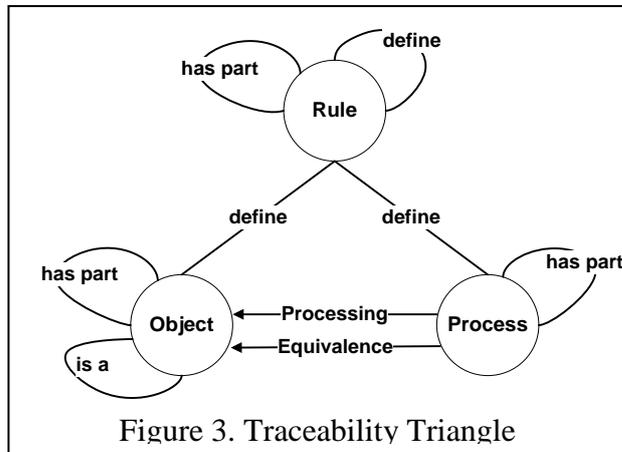


Figure 3. Traceability Triangle

three basic categories and five relationships used to describe a system. Only two of these categories are primary units of estimation – *Process* and *Object*. *Rule* is introduced to facilitate change propagation because of changes of processes and objects more often are initiated from rules changes.

More precise definitions of these categories can be found

in [8].

The relationships “has part” and “is a” allow to produce descriptions of desired granularity. It’s the key point to manageable cost on traceability maintenance.

Despite of minimal set of categories and a high level of abstraction the introduced traceability triangle allows constructing rules to control requirements errors. For example we can define the rule “«there must exist a process for any object »”, which allows controlling errors on the higher level of system description.

Due to selected set of categories TT allows to simply integrate any design methods to conduct change propagation through all design models produced in a project. It is based on that any design model describes processes or objects of the developed software system because this is the ultimate goal of design - to identify the objects and processes that make up the system. Moreover if an analyst could not say with which processes or objects of a system his models are interrelated he must rethink his work.

As introduced TT allows upper-level integration of design methods any alterations of the latter will not destroy the planning system and will require less time to reconfigure the overall traceability.

The plain categories and relationships allow reducing time and errors during a non automatic process of relating design artifacts to the traceability model, decreasing semantic errors which cannot be automatically discovered.

It should be noted that the integration method based on TT does not cancel studies on peer-to-peer integration of design methods, such as BPMN-to-UML, or UML, they can be successfully used in conjunction providing different levels of traceability for various projects tasks.

TT is a basic model which allows developing not so much of the axioms

for errors checking. To extend the set of axioms and allow grouping and layering of a large amount of design data which is usually produced during Enterprise system development we construct the extension of TT for this domain. This extension is the three-level balanced ontological model is presented at the Fig 4. This model is the reflection of TT for the main design stages of Enterprise system and it corresponds to 1-4th layers of Zachman Framework [4].

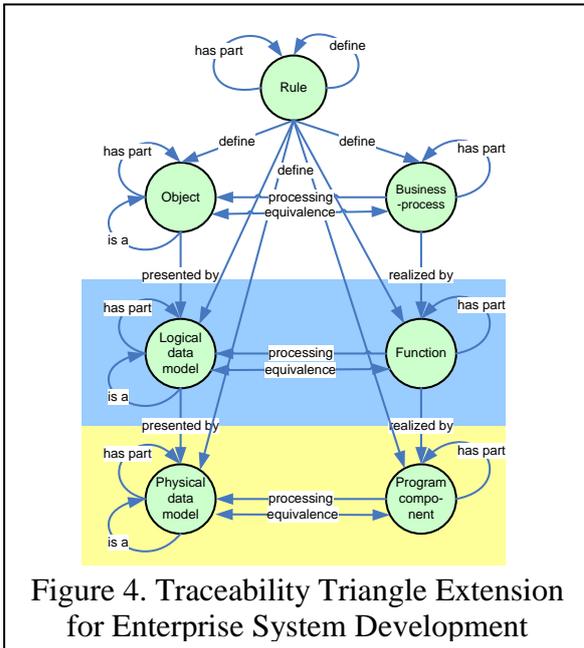


Figure 4. Traceability Triangle Extension for Enterprise System Development

industrial enterprises in the field of cooperation of technological chains inside the cluster. The proposed solution was allowing revealing errors at early design stages raising reliability of planning and reducing expenses of modification.

CONCLUSION

All solutions of the developed complex have been tested in several projects. One of such projects was the integration of

REFERENCES

- [1] A. Kannenber and H. Saiedian, "Why Software Requirements Traceability Remains a Challenge," Crosstalk, (July), 2009, pp.14-19.
- [2] R. Oliveto, A. Marcus, J. Huffman Hayes, "Software Artefact Traceability: the Never-Ending Challenge," ICSM 2007, pp.485-488.
- [3] CHAOS Summary 2009, Standish Group, report, 2009.
- [4] Zachman J. A., Sowa J. F, Extending and formalizing the framework for information systems architecture // IBM Systems Journal. 1992, vol. 31, № 3, P. 590-561.
- [5] Assawamekin N., "An Ontology-Based Approach for Multiperspective Requirements Traceability between Analysis Models," Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on , vol., no., pp.673-678, 2010.
- [6] Мадорская Ю.М., Тимофеев А.Н. О проблеме образования в области инженерии программного обеспечения // Международная научно-техническая конференция «Наука и образование». – Мурманск. – 2011.- С.203-207
- [7] Мадорская Ю.М. Метод оценки сложности модификации программного обеспечения АСУП: автореф. дис. канд. техн. наук / Ю.М. Мадорская. – СПб.: СПбГПУ, 2011. –18 с.
- [8] Мадорская Ю.М. Метод оценки сложности модификации программного обеспечения АСУП: дис. канд. техн. наук / Ю.М. Мадорская. – СПб.: СПбГПУ, 2011. –119 с.