A Traceability-based Method to Support Conceptual Model Evolution

Marcela Ruiz

PROS Research Centre, Universitat Politècnica de València, Spain
lruiz@pros.upv.es

Abstract. Renewing software systems is one of the most cost-effective ways to protect software investment, which saves time, money and ensures uninterrupted access to technical support and product upgrades. There are several motivations to promote investment and scientific effort for specifying systems by means of conceptual models and supporting its evolution. As an example, the software engineering community is addressing solutions for supporting model traceability, continuous improvement of business process, organisational reengineering, information system maintenance, etc. Model-driven techniques have been developed in order to analyse systems raising the abstraction level of its specification. However, a support for conceptual model evolution by means of model-driven techniques is still needed. This thesis proposes a traceability-based method that involves model-driven capabilities for designing and providing guidelines, techniques, and tools to support conceptual model evolution. The main idea is to support information system analysts in the tasks related to: justify **why** the conceptual models have evolved, report and specify **what** elements have evolved, and guide **how** to carry out evolution in certain predefined organisational contexts. We plan to apply our method to guide the evolution of an E-shopping software. This way, we also provide mechanism to facilitate industrial adoption.

Keywords: conceptual model evolution, reengineering frameworks, traceability-based support, business process modelling, intentional modelling, pattern definition, delta analysis

1 Introduction

Software maintenance and information system evolution are activities that receive significant dedication by industry. This is one of the reasons that motivate the information systems engineering community to investigate in this area. Organisations are aware on the need to apply mechanisms and strategies in order to encompass processes and products in changing environments. For instance, in organisational context, companies need to rethink business processes, infrastructures, technologies, resources, etc. according to new demands from their environment or changes in their organisational objectives. Business processes should also be transformed to support the new processes and tasks that result from the involvement of new objectives or goals in the organisation. Then, constant organisational change and its influence in processes and products must be considered as a fundamental rule of competitive strategy for continuous improvement [1]. For software systems, the high pressure of a very short time-to-market often forces developers to implement the code of the application directly, without using a disciplined development process, which may have disastrous effects on the quality and documentation of the delivered software application [2]. These practices have been the motivation for opening new research lines in order to support post-delivery life-cycle activities. Besides, with regard to the keynote of the ERCIM News 88 magazine[1], some of external drivers for changing software are innovation, cost reduction and regulation; factors that need to be supported by techniques, tools and methods.

The main goal of my PhD thesis is *to design a traceability-based method that involves model-driven capabilities in order to support conceptual model evolution*. The main idea is to provide a model-driven method that can be used by information system analysts in order to provide them with reports and evidences to help decision making in information system evolution contexts. This paper summarizes the author's PhD work and project, working for two years and a half, under the supervision of Dr. Sergio España Cubillo in the PROS Research Centre of the Universitat Politècnica de València.

2 Problem Description and Research Methodology

Traditionally in software system development, the evolution process and information system maintenance have been faced by means of the reengineering process, change specification, evolution metrics, goal-driven requirements engineering and model management. For these reason, we explore current solutions in these fields in order to find related research that confronts conceptual model evolution.

The reengineering process is commonly defined and widely used by the scientific community by means of the metaphor of the "horseshoe" model, which purpose is to present the reengineering process in a figure (the horseshoe is basically a left-hand side, a right-hand side and a bridge between the sides). In general terms, the left-hand side of the horseshoe model consists of an extraction from an existing system to get the system specification, the right-hand side consist of conventional software development activities, and the bridge between the sides consists of a set of transformations from the old system to the new one [4]. Both, the left-hand side and right-hand side represent different levels of abstraction of the system. Nowadays, the Object Management Group (OMG) is working on promote an industrial consensus on modernisa-

[1] The ERCIM News 88 special theme was "Evolving Software" 3. Visser, J., *Change is the constant*, in *ERCIM news - Special theme: Evolving Software*. 2012: Sophia Antipolis Cedex, France. p. 3.. The magazine put together a set of papers to give an overview of both traditional and emerging software engineering techniques, tools and approaches used by software evolution experts.

tion of existing application by means of the initiative named Architecture-Driven Modernisation (ADM) [5]. This initiative is based on the MDD paradigm to automate the horseshoe model. However, full support for the evolution process (the bridge between the sides) is still missing. The authors of [6] aimed to automate the horseshoe model, although it is not severely applied.

Goal-driven requirements engineering approaches faced goal modelling from different perspectives of use. Some of those uses are: understanding the current organisational situations and need for change, decision making, relating business goals to functional and non-functional system components and validation of compliance between system specification and stakeholders' goals [7]. Co-evolution approaches has been proposed in order to understand reciprocal evolution of system components [8]. Nevertheless goal specification related with change models and specification of evolution grains is still an open research field.

System change and stability analysis in order to derive or facilitate system evolution is confronted by [9]. A method to support the elicitation of evolution requirements and a generic syntax to specify them is explored in [10]. Also, metrics for classifying and measuring software evolution are analysed by [11]. Even though, specification of evolution in with formal conceptual models and measurement techniques to provide meaningful to kick start analysis is still needed.

Model management confront problems in many databases application domains (e.g. data warehousing, semantic query processing, meta-data management, meta-data integration, schema evolution etc.); research projects in this area are aiming at providing high-level abstractions artefacts in order to offer a generic solution [12-13]. Bernstein [14] presents a full description of all of the model management operators. Moreover, no complete frameworks to support enterprise information system evolution have been proposed yet.

The problems detected establish the motivations in which this PhD thesis is founded.

2.1 Research Questions Objectives and Means

We follow design science to classify our research questions in knowledge problems (KP) and practical problems (PP) [15]. This way, we are looking for highlighting our research results by means of producing useful artefacts. This thesis is focused on conceptual model evolution. To achieve the main goal, we conceive the following research questions:

- **RQ1** (KP). What elements are common in conceptual model evolution? The answer to this question should clarify terminology, stakeholders, and helps to establish a conceptual framework to facilitate reasoning about conceptual model evolution.

- **RQ2** (KP). Which are the current conceptual model evolution methods? The answer to this question should establish the state of the art about current conceptual model evolution support.
  - **RQ2.1** (KP). Which of these methods are model-driven oriented?
- **RQ3** (PP). How can be supported a conceptual model evolution method? The answer to this question refers to the main **goal** of this thesis.
  - **RQ3.1** (PP). What guidelines are needed in order to evolve conceptual models?
  - **RQ3.2** (PP). What techniques are needed in order to facilitate the use of the method?
  - **RQ3.3** (PP). What tools are needed in order to support the use of guidelines and techniques?
- **RQ4** (PP). How can possible scenarios be integrated in the conceptual model evolution method? The answer to this question refers the modules to support business process evolution, goal-driven evolution, and reengineering.
- **RQ5** (KP). How can the model-driven method to support conceptual model evolution be validated? The answer to this question should establish a validation framework to measure feasibility, trade-off and sensitivity.

Means

To achieve the main goal and solve the research questions, three main means are conceived: a) Expert views. My directors are experts to guide my decisions to provide solutions of the addressed problem. b) Technological support. We are expert in model-driven tools as Eclipse. This way, we have capabilities to provide tool support for the method. c) Collaboration with other research groups. Collaboration increases our perspectives to provide solutions. d) Action research. Our proposal is motivated by the needs of real information system analysts.

3 Research Methodology

This PhD project follows the design science framework to design a new artefact: a model-driven method to support conceptual model evolution. The research methodology is explained by means of regulative cycles that were conceived in order to answer the research questions. Fig. 1 presents the research methodology.

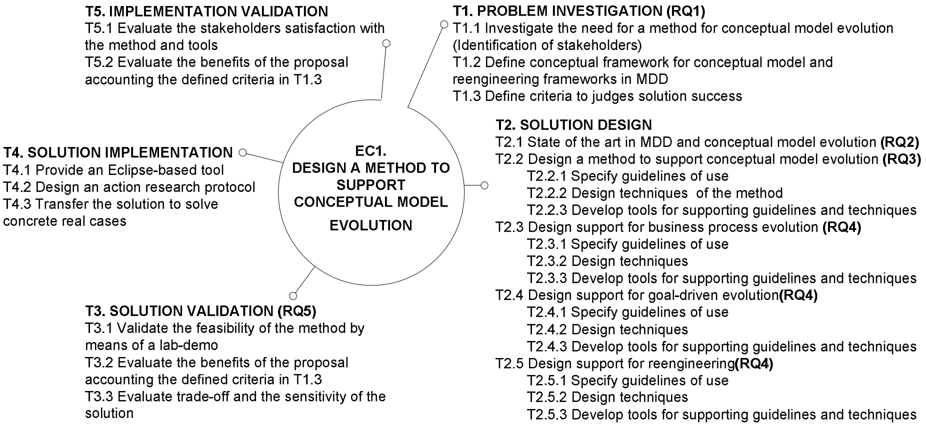
T5. IMPLEMENTATION VALIDATION
T5.1 Evaluate the stakeholders satisfaction with the method and tools
T5.2 Evaluate the benefits of the proposal accounting the defined criteria in T1.3
T1. PROBLEM INVESTIGATION (RQ1)
T1.1 Investigate the need for a method for conceptual model evolution (Identification of stakeholders)
T1.2 Define conceptual framework for conceptual model and reengineering frameworks in MDD
T1.3 Define criteria to judges solution success
T4. SOLUTION IMPLEMENTATION
T4.1 Provide an Eclipse-based tool
T4.2 Design an action research protocol
T4.3 Transfer the solution to solve concrete real cases
EC1.
DESIGN A METHOD TO SUPPORT CONCEPTUAL MODEL EVOLUTION
T2. SOLUTION DESIGN
T2.1 State of the art in MDD and conceptual model evolution (RQ2)
T2.2 Design a method to support conceptual model evolution (RQ3)
T2.2.1 Specify guidelines of use
T2.2.2 Design techniques of the method
T2.2.3 Develop tools for supporting guidelines and techniques
T2.3 Design support for business process evolution (RQ4)
T2.3.1 Specify guidelines of use
T2.3.2 Design techniques
T2.3.3 Develop tools for supporting guidelines and techniques
T2.4 Design support for goal-driven evolution(RQ4)
T2.4.1 Specify guidelines of use
T2.4.2 Design techniques
T2.4.3 Develop tools for supporting guidelines and techniques
T2.5 Design support for reengineering(RQ4)
T2.5.1 Specify guidelines of use
T2.5.2 Design techniques
T2.5.3 Develop tools for supporting guidelines and techniques
T3. SOLUTION VALIDATION (RQ5)
T3.1 Validate the feasibility of the method by means of a lab-demo
T3.2 Evaluate the benefits of the proposal accounting the defined criteria in T1.3
T3.3 Evaluate trade-off and the sensitivity of the solution

Fig. 1. Overview of the research methodology

Since our proposal focus on the development of a new artefact, the main cycle of the research methodology is an engineering cycle (EC1. Design a model driven method to support data system evolution). Concretely, this cycle is formed by 5 main tasks: T1) problem investigation; T2) solution design; T3) solution validation; T4) solution implementation; and T5) implementation validation.

An information system needs evolve. Since the information system is specified by means of models, we investigate current research to support conceptual model evolution. We identify the stakeholders or possible users of the method. To define the problem and define the method, we provide a conceptual framework to avoid terminology incoherence. In addition, we establish the criteria to judge the solution success when we finish the engineering cycle. These activities are related to T1.

In T2 we explore available solutions by reviewing state of the art. We design a new solution; i.e. our method. To do that, we design the guidelines of use; we provide techniques to facilitate the use of the method; and we develop tools (prototypes built in the laboratory) to support guidelines and techniques. Also, we design the support for the modules of business process evolution, goal-driven evolution and reengineering frameworks.

The method is validated in T3. We demonstrate the feasibility by means of lab-demo. We establish a comparative with the results of the lab-demo with the criteria defined in T1.3. Also, we evaluate trade-off and sensitivity of the solution.

In T4 we implement the method using Eclipse based tools, design an action research protocol to transfer the solution to be used in practice. Finally, in T5 we assess the operability of the tool, stakeholder's satisfaction and criteria of success by means the results of the action research protocol carried out in T4.

4 Proposal

We face the design of the method by two main motivations: 1) Market pull or demand pull and 2) Technology push [16]. The first one refers our motivation to evolve the E-Shopping software (a real case and we have into account the user needs). We call it market-driven solution. The second one refers our motivation to provide an invention without proper consideration of whether or not it satisfies a set of specific user needs. We call it technology push-driven solution.

To design the method, we have been inspired by the metaphor of a “horseshoe” of Kazman et. al. [4]. Carrying the horseshoe metaphor to the MDD field, an interesting evolution method can be provided for different scenarios. As a result, models are the main artefact and the analysis of them is in a high level of abstraction. The traceability-based support plays the main role in the method; it provides two types of traces: Vertical traces to relate elements that specify different characteristics of information systems (e.g., processes, goals, etc.); and horizontal traces are accounted to relate evolution of elements.

To use the method, the analyst should carry out the four tasks presented in the Fig. 2:

1. Define evolution question, in this task the analyst decides what characteristic of evolution process want to know. The analyst follows a set of guidelines in order to know if s/he wants to obtain information about justifying **why** the conceptual models have evolved, reporting or specifying **what** have evolved, or analysing **how** to evolve conceptual models according to a set of predefined solutions for certain contexts.

2. Specify As-Is and To-Be models, in this task the analyst specify the current and desired system to be analysed applying the evolution modules.

3. Apply evolution modules, in this task the analyst applies the module that corresponds with s/he evolution question.

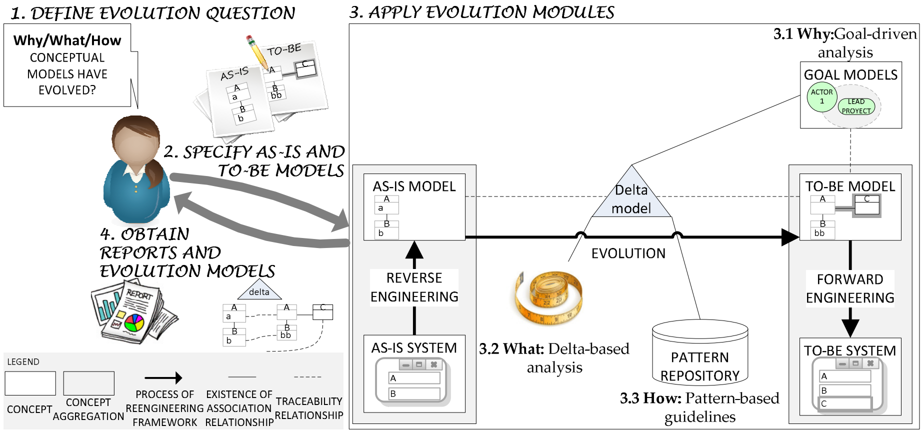
1. DEFINE EVOLUTION QUESTION
Why/What/How CONCEPTUAL MODELS HAVE EVOLVED?
AS-IS
TO-BE
2. SPECIFY AS-IS AND TO-BE MODELS
4. OBTAIN REPORTS AND EVOLUTION MODELS
REPORT
delta
3. APPLY EVOLUTION MODULES
3.1 Why:Goal-driven analysis
GOAL MODELS
ACTOR 1
LEAD PROYECT
AS-IS MODEL
Delta model
TO-BE MODEL
EVOLUTION
REVERSE ENGINEERING
FORWARD ENGINEERING
AS-IS SYSTEM
3.2 What: Delta-based analysis
PATTERN REPOSITORY
3.3 How: Pattern-based guidelines
TO-BE SYSTEM
LEGEND
CONCEPT
CONCEPT AGGREGATION
PROCESS OF REENGINEERING FRAMEWORK
EXISTENCE OF ASSOCIATION RELATIONSHIP
TRACEABILITY RELATIONSHIP

Fig. 2. Overview of the traceability-based Method to support conceptual model evolution

For the **why** question (*3.1 Why: Goal-driven analysis*), a goal driven model evolution support is provided. The vertical traceability is established between two information system specification languages. As a proof of concept, we have aligned the *i** framework with the Communication Analysis modelling techniques. Goal models are connected with delta models that specify changes in the information system.

For the **what** question (*3.2 What: Delta-based analysis*), a set of metrics are provided in order to report meaningful information about the evolution processes, elements involved and the conceptual impact of changes.

For the **how** question (*3.3 How: Pattern-based guidelines*), a set of patterns to evolve business process models have been established. The patterns are connected with delta models to register what changes implies the application of patterns.

4. Obtain reports and evolution models, in this task the analyst obtain the results of modules application. Based on the results the analyst can provide meaningful information about conceptual model evolution processes and make decisions based on evidences.

The method is in continuous improvement and re-adjusts. The modules have been designed; the implementation has being developed in Eclipse-based tools.

5 Progress of the Thesis

In 2012, organisational reengineering frameworks have been studied, focusing on RQ1 and RQ2. Furthermore, the alignment between the process and the goal perspectives were explored. As a proof of concept, we have aligned the *i** framework with the Communication Analysis modelling techniques. This proof of concept refers the RQ4. Also, we implemented the alignment of this modelling languages in an Eclipse-based tool (this implementation refers RQ3.). And we analysed the benefits and the limitations of aligning process and goal perspectives. We started a first version of the definition of the artefacts to support model evolution (Traceability support).

In 2013, the modules of the method were designed and reported. We carried out an experimental task with master students to analyse vertical traceability between conceptual models.

In 2014-2015 we plan to establish the method guidelines and delta analysis technique formalisation. In addition, we are looking for implementing pattern definition metamodel and evolution metamodel in an Eclipse plug-in (RQ3).We plan to validate the method and the prototype by means of laboratory demos. The idea is to estimate scalability, trade-off and sensitivity of our method. This validation refers RQ5.

We plan to finalize the implementation and the implementation validation of the method in 2015.

Acknowledgments
I acknowledge to my supervisor Sergio España for his invaluable support and advices to drive my thesis and encourage my research career.

This PhD project has been supported by the Spanish Generalitat Valenciana ORCA (PROMETEO/2009/015); the FPI grant of the Universitat Politècnica de València (3146); the European Commission FP7 Project CaaS (611351); and the ERDF structural funds.

References

1. Sage, A.P., *Systems Engineering and Systems Management for Reengineering.* Systems Software, 1995(30): p. 3-25.
2. Di Lucca, G.A., A.R. Fasolino, and P. Tramontana, *Reverse engineering Web applications: the WARE approach.* Journal of software maintenance and evolution: Research and practice, 2004. **16**(1-2): p. 71-101.
3. Visser, J., *Change is the constant*, in *ERCIM news - Special theme: Evolving Software.* 2012: Sophia Antipolis Cedex, France. p. 3.
4. Kazman, R., S.G. Woods, and J.S. Carrière, *Requirements for Integrating Software Architecture and Reengineering Models: CORUM II*, in *Working Conference on Reverse Engineering (WCRE 1998).* 1998.
5. OMG. *Architecture-Driven Modernization (ADM).* 2012; Available from: http://adm.omg.org/.
6. Sánchez Cuadrado, J., et al., *Parametrización de las transformaciones horizontales en el modelo de herradura*, in *Jornadas de Ingeniería de Software y Bases de Datos (JISBD'12).* 2012: Almería, Spain.
7. Kavakli, E. and P. Loucopoulos, *Goal Driven Requirements Engineering: Evaluation of Current Methods*, in *Exploring Modelling Methods for Systems Analysis and Design (EMMSAD 2003).* 2003: Klagenfurt/ Austria.
8. Etien, A. and C. Salinesi. *Managing requirements in a co-evolution context.* in *Requirements Engineering (RE 2005).* 2005. Paris, France.
9. Herrmann, A., A. Wallnöfer, and B. Paech, *Specifying changes only - a case study on delta requirements*, in *REFSQ 2009.* 2009, Springer-Verlag: Essen, Germany. p. 45-58.
10. Salinesi, C., A. Etien, and I. Zoukar, *A systematic approach to express IS evolution requirements using gap modelling and symilarity modelling techniques*, in *International Conference on Advanced Information Systems Engineering (CAiSE'05).* 2005: Porto, Portugal.
11. Etien, A. and C. Rolland, *Measuring the fitness relationship.* Requirements Engineering Journal, 2005(10): p. 184-197.
12. Rahn, E. and P.A. Bernstein, *A survey of approaches to automatic schema matching.* VLDB 2001. **10**: p. 334-350.
13. Madhavan, J., P.A. Bernstein, and E. Rahn, *Generic Schema Matching with Cupid*, in *27 th International Conference on Very Large Data Bases (VLDB 2001).* 2001: Roma, Italy.
14. Bernstein, P.A. *Applying Model Management to Classical Meta Data Problems.* in *First Biennial Conference on Innovative Data Systems Research (CIDR 2003).* 2003. Asilomar.
15. Wieringa, R., *Design Science as Nested Problem Solving*, in *4th International Conference In Design Science Research In Information System and Technology (DESRIST'09).* 2009: Malvern, PA, USA.
16. Martin, M.J.C., *Managing Innovation and Entrepreneurship in Technology-Based Firms*, ed. I.T.O.E. MANAGEMENT. Vol. 43. 1996.