Testing-Based Conceptual Schema Validation in a Model-Driven Environment

Maria Fernanda Granda

Department of Information Systems and Computation, Universitat Politècnica de València, Valencia, Spain

mgranda@dsic.upv.es

Faculty of Engineering, University of Cuenca, Cuenca, Ecuador

fernanda.granda@ucuenca.edu.ec

Abstract. In Model-Driven Development, the main artefacts are conceptual schemas, and efforts are focused on their creation, testing and evolution at different levels of abstraction through transformations. If a conceptual schema has defects, these are passed on to the following stages, including coding. Therefore, techniques for improving the quality of conceptual schemas must be implemented to assure the correct generation of final software products. One of the challenges in Model-Driven Development is being able to identify defects early on, at the level of conceptual schemas, as this would help reduce development costs and improve software quality. In this research proposal we suggest an approach for testing-based conceptual schema validation in order to improve quality.

Keywords: early testing, conceptual schemas, Model-Driven Development, validation & verification.

1 Introduction

Despite much scepticism and many problems [1], Model-Driven Development (MDD) is being used and improved in order to provide multiple many inherent potential benefits for industry [2], [3]. One of its greatest benefits is the ability to handle the complexity of software development by raising the abstraction level. Models are expressed using concepts that are not related to a specific implementation technology, which means that the models can be easier to specify, understand, maintain and document. As in Model-Driven Engineering (MDE), the primary artefacts are the conceptual models, and ensuring their quality at an optimum level is still challenging for researchers and developers.

Although verification[1] and validation[2] (V&V) are highly related to the concepts of quality and software quality assurance, very few MDD tools incorporate these

[1] Verification is to check that the conceptual schema meets its stated functional and non-functional requirements [20].

Doctoral Consortium of CAiSE, 2013.

activities into their development process. The OO-Method (OOM) [4], a Model Driven Architecture (MDA) approach, is a model-driven initiative with a technical multi-view (structural model, dynamic model, functional model and presentation model), where the structural view is the basis for the automatic derivation of the other views, and this feature helps to minimize the problems such as multi-view specifications and synchronization, integration and change propagation. The OO-Method has been successfully implemented in industry through the Integranova[3] commercial tool (previously known as OLIVANOVA). This tool manages the syntactic verification of conceptual schemas (e.g. syntactic correctness) [4], but it still does not validate whether the model built meets the requirements and expectations of the stakeholders.

With the ever-increasing complexity of software systems, the ability to identify the vast majority of defects early on at the model level is a challenge that if met could help to reduce development costs and improve software quality [5]. However, to assess the quality of a conceptual schema, we need a quality model. In the literature, we can find several proposals (e.g. Mohagheghi et al [6], Krogstie [7]). We will aim to set the quality properties that can be improved using testing techniques.

Testing is part of a process of V&V, where the conceptual schema operates under controlled conditions, (1) to verify that it behaves as specified; (2) to detect defects, and (3) to validate user requirements [29]. Lightweight testing techniques of the conceptual schemas are required to locate and point out defects in realistic schemas with minimum cost.

This work aims to define a testing-based validation technique for multi-view conceptual schemas (i.e. structural and behavioural). We will focus on adapting testing techniques for MDD environments, such as the OO-Method approach, because we believe that testing can be a very effective and efficient way to identify defects early on, and can play an important role in the validation of conceptual schemas.

The paper is structured as follows: the next section summarizes related work. Then, in Section 3, we present the problem statement and explain the research questions that we aim to answer in the proposed work. Section 4 describes the research methodology to be applied. Section 5 presents an overview of our solution design proposal. And finally, section 6 presents the conclusions of this work.

---

[2] Validation is to ensure that the conceptual schema meets the customer's expectations [20].

[3] http://www.integranova.com/

2 Related Work

From a mapping study we found 161 studies on V&V techniques for conceptual schemas. However, we found only 4 approaches[4] which apply testing at the modelling level [8], [9], [10] and [11].

Table 1 shows these testing techniques for the information system domain that have been proposed in the last ten years. The modelling language most used is the standard Unified Modelling Language (UML). Therefore, the analysis is performed on structural (S) and behavioural (B) models. In this table, we can also see the similarities and differences of our proposal when compared to the related work.

Table 1. Techniques for testing conceptual schemas

Feature /Author		[8]	[9]	[10]	[11]	Our Proposal
Conceptual Schema Under Test	Model	S	S, B	B	S, B	S, B
	Integrated MDD	No	No	possible use	possible use	Yes
	Generation Mode	Manual (CTSL)	Automatic (TAM)	Not required	Automatic (JAL)	Automatic
	Integrity Constraint supported?	No	Yes	No	No	Yes
Quality	Focus	Correctness, Completeness	Consistency multi-view	Debuggable Testable	Consistency constraints	6C goals[6]
Testing Technique Features	Test Source	Requirements document	Conceptual Schema	Conceptual Schema	Conceptual Schema	Requirements model
	Generating Test	Manual	Automatic	Automatic	Manual	Automatic and Manual
	Generating Test Data	Manual	Semi-automatic	Manual	Manual	Automatic and Manual
	Coverage Criteria?	Yes	Yes	No	Yes	Yes
	Repairing Feedback?	No	Yes	Yes	Yes	Yes

Tort's approach [8] is the closest to our work, because it focuses on validating the UML conceptual schema with respect to the stakeholder's requirements. However, this solution requires the development of certain skills in order to enter sequences of

4 Articles on the same technique and by same author are considered as a single approach.

test cases using Conceptual Schema Testing Language (CSTL), which makes this method unsuitable for a MDD environment.

The remaining two testing work that proposed by Pilskalns [9] and Dinh-Trong et al [11]. These test the consistency between the UML's diagrams including OCL expressions. Pilskalns's approach is based around constructing a Testable Aggregate Model (TAM). Dinh-Trong et al describe the actions using Java-like Action Language (JAL). Dotan and Kirshin [10] have a plug-in for IBM Rational modelling tools, allowing the execution, debugging and testing of UML models (activity diagrams and state machines) through animation. However, these last three do not focus on validating the conceptual schema with respect to the stakeholder's requirements.

It is important to indicate that an important initiative for building executable UML models is the fUML [13], which is promoted by the OMG (Object Management Group). An ongoing research on a model execution framework based on fUML is presented by Mayerhofer [14]. This framework will enable efficient testing and validating of UML models by providing debugging capabilities, as well as a test.

In contrast to these testing techniques, a greater number of V&V techniques for conceptual models (e.g. [15], [12], [16]) were also found, but they do not use any testing strategy. They are focused on the study of the desirable properties of conceptual schemas (e.g. a well-formed instantiation of the model, and consistency between models and with constraints) and the development of automated reasoning procedures or the semi-automated control of them.

3 Problem Statement and Research Questions

Requirements errors are the most common cause of defects in system development projects [17]. This suggests that it would be more effective to focus quality assurance efforts on early phases, in order to catch defects as soon as they occur. In MDD, the ability to identify defects early on is still a challenge that, if it were met, could help to reduce development costs and improve the quality of delivered software systems. Lightweight testing techniques for improving the quality of the conceptual schemas must be implemented. These techniques should be able to find defects with minimum effort, and without the need for a strong testing background.

Our work aims to define a testing-based validation approach to improve the quality of conceptual schemas built in an MDD environment.

In order to achieve our objective, we have identified the following research questions (RQs).

RQ_1: How can conceptual schemas be validated by using testing techniques in an MDD environment?

- $RQ_{1.1}$: Which testing techniques can be effectively used or adapted at the conceptual model level?
- $RQ_{1.2}$: What kind of defects can be detected at the conceptual modelling level using a testing strategy?
- $RQ_{1.3}$: Which MDD environment requirements should be considered when developing the testing-based approach?
- $RQ_{1.4:}$ How can an approach for testing-based validation of conceptual schemas be integrated into an MDD environment?

RQ_2: To what extent will our testing-based validation approach contribute to ensuring the quality of conceptual schemas?

- $RQ_{2.1}$: Which existing quality assurance frameworks can be more suitable for use in MDD environments?
- $RQ_{2.2}$: What quality properties can be improved using testing techniques in conceptual schemas?

In the next these questions are presented in more detail within a research cycle proposed by Wieringa in order to structure the research methodology [18].

4 Research Methodology

The type of research methodology used corresponds to design science framework since its purpose is the design of a new testing-based validation approach and it is therefore a matter of design [19]. We follow the regulatory cycle proposed by Wieringa [18]. This cycle can be instantiated in two forms: the engineering cycle and the research cycle. We will cover the first three phases of the engineering cycle[5], and some phases will be embedded into research cycles[6]. The Engineering cycle is rational problem decomposition for the practical problem. It starts with a ***problem investigation*** phase, where we seek to understand how to validate and verify conceptual schemas by using testing techniques, and what current approaches have been proposed to achieve this. To do this, based on existing surveys and systematic reviews concerning software testing, we select some testing strategies (RQ1.1, RQ1.2) as possible candidates to implement our approach. We also identify the most

[5] *Engineering cycle* (problem investigation, solution design, design validation, design implementation and implementation evaluation) is associated with practical problems. However, practical problems may contain knowledge questions [18].

[6] *Research cycle* (research problem investigation, research design, research design validation, research, analysis of results) is associated with knowledge problems. However, knowledge problems may contain practical questions [18].

relevant quality properties which need to be considered for conceptual schemas built in an MDD environment (RQ2.1, RQ2.2); as well as the characteristics and resources needed to support the testing-based validation approach (RQ1.3). By considering these identified quality properties, we can analyze and identify which properties are affected by defects that have been detected in conceptual models so far. So, based on the relationships between quality properties and defects, we can evaluate the selected testing techniques (RQ1.1) in order to better understand which ones can be effectively used or adapted for our purpose. One of the outcomes of this phase will be a conceptual framework that should aid our understanding of the proposed approach.

The next phase is ***Solution Design***, which is characterized by its iterative nature. This phase solves a practical problem by designing a testing-based validation method that can be embedded into a MDD environment (RQ1.4). Our initial solution proposal is based on design specifications identified in the previous phase, and our own logical reasoning. The next iteration refines the solution by adding insights from several interviews with experts from academia and industry to improve the approach. Further iterations use input from the analysis of several laboratory demonstrations. Finally, the ***Solution Validation*** phase solves a knowledge problem which asks if the solution design (prototype) is effective and efficient (e.g. finding defects capability, functional coverage). We then validate our approach by conducting various experiments in order to answer RQ2.

Table 2 shows the general and specific objectives for each research question. We distinguish between knowledge problems (KP) and practical problems [7] (PP), as defined by Wieringa [18]. The table includes an estimated timescale for achieving these objectives.

Table 2. Objectives of Research Proposal

RQ	Objectives	Time(months)
RQ_1 (PP)	**Develop an approach for validating conceptual schemas using testing techniques.**	10
$RQ_{1.1}$ (KP)	Identify and analyze testing techniques that can be used at the conceptual model level.	3
$RQ_{1.2}$ (KP)	Identify categories of defects, with respect to the degree of coverage that they can have on conceptual schemas.	3
$RQ_{1.3}$ (KP)	Identify characteristics and resources needed as support for the testing-based validation approach.	1
$RQ_{1.4}$ (KP)	Integrate the approach proposed into an MDD environment.	8
RQ_2 (KP)	**Assess the contribution of the testing-based approach in ensuring the quality of conceptual schemas.**	8
$RQ_{2.1}$ (KP)	Identify the quality assurance framework that could be suitable for use in MDD environments.	1
$RQ_{2.2}$ (KP)	Identify the most significant quality properties that can be improved using testing techniques in conceptual schemas.	2
	Total	**36 months**

[7] A *knowledge problem* is a lack of knowledge about the world (as long as someone desires to fill this gap), whereas a *design problem* (a.k.a. practical problem) is the difference between the way the world is and the way someone thinks it should be.

5 Solution Design Proposal

Figure 1 shows an overview of our solution design, which is based on a series of models transformations for automatically generating test cases from a requirements model. These test cases will be used for testing the conceptual schema, previously prepared for use as a testing artefact (conceptual schema under test).

With regard to the results achieved so far, a systematic mapping study has enabled definition of a defects taxonomy for conceptual schemas. We are currently in the validation phase of this taxonomy. This outcome will assist in answering RQ 1.2. Moreover, with the purpose of investigating the feasibility of existing languages and tools for executing models (RQ 1.3), we have started exploring the USE tool that allow the verification of OCLs on conceptual models, and the CSTL language, proposed by Tort [8] in specific conceptual schemas generated with the INTEGRANOVA tool.

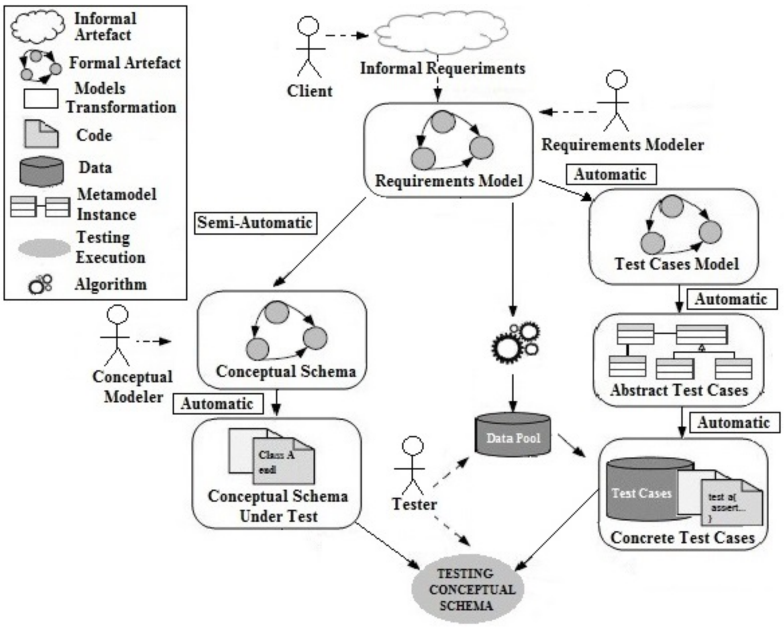

Fig. 1. Early-testing approach for conceptual schemas

The contribution to be expected from this proposed work is the testing-based validation of conceptual schemas, by automatically generating test cases from requirements models (specifications). In addition, the integration of our approach to an existing quality assurance framework for MDD environments (e.g. [6]) will improve support for decision-making in the prioritising of repair of defects detected at the conceptual schema level.

Acknowledgements

My deepest gratitude goes to my supervisors Nelly Condori-Fernández, Oscar Pastor and Tanja Vos for their unfailing support and invaluable comments.

References

1. Franc, R., Rumpe, B.: Model-driven Development of Complex Software: A Research Roadmap. In : International Conference on Software Engineering, IEEE, DC (2007)
2. Staron, M.: Adopting Model Driven Software Development in Industry – A Case Study at Two Companies. Model Driven Engineering Languages and Systems 4199, 57-72 (2006)
3. Hutchinson, J., Rouncefield, M., Whittle, J.: Model-driven engineering practices in industry. Proceedings of the 33rd International Conference on Soft. Eng., 633-642 (2011)
4. Pastor, O., Molina, J.: Model-Driven Architecture in Practice. A Software Production Environment Based on Conceptual Modeling. Springer, New York (2007)
5. Van Der Straeten, R., Mens, T., Van Baelen, S.: Challenges in Model-Driven Software. In : Models in Software Engineering. Springer, Tolouse, Fr. (2009) 35-47
6. Mohagheghi, P., Dehlen, V., Neple, T.: Definitions and approaches to model quality in model-based software development– A review of literature. Information and Software Technology 51, 1646-1669 (2009)
7. Krogstie, J.: Model-Based Development and Evolution of Information Systems. A Quality Approach. Springer, London (2012)
8. Tort, A., Olivé, A., Sancho, M. R.: On Checking Executable Conceptual Schema Validity by Testing. In : 23rd International Conference, DEXA, Viena, Austria, pp.249-264 (2012)
9. Pilskalns, O., Andrews, A., Knight, A., Ghosh, S., R., F.: Testing UML designs. In : Information and Software Technology, MA, USA (2007)
10. Dotan, D., Kirshin, A.: Debugging and testing behavioral UML models. In : Proced. of the Conference on O-Oriented Programming Systems, Languages, and Applications (2007)
11. Dinh-Trong, T., Kawane, N., Ghosh, S., France, R., Andrews, A. A.: A tool-supported approach to testing UML design models. In : Proc. of the ICECCS, CO, USA (2005)
12. OMG: Semantics of a Foundational Subset for Executable UML Models (FUML).
13. Mayerhofer, T.: Testing and Debugging UML Models Based on fUML. In Society, I., ed. : ICSE 2012. Doctoral Symposium, Zurich, Switzerland, pp.1579-1582 (2012)
14. Gogolla, M., Kuhlmann, M., Hamann, L.: Consistency, independence and consequences in UML and OCL models. Lecture Notes in Computer Science 5668 LNCS, 90-104 (2009)
15. Queralt, A., Teniente, E.: Verification and validation of UML conceptual schemas with OCL constraints. ACM Transactions on Software Engineering and Methodology (2012)
16. Bergmann, G., Hegedüs, A., Horváth, A., Ráth, I., Ujhelyi, Z., Varró, D.: Implementing Efficient Model Validation. 2011 26th IEEE/ACM International Conference on Automated Software Engineering, ASE 2011, Proceedings(6100130), 580-583 (2011)
17. El Emam, K., Günes, A. G.: A Replicated Survey of IT Software Project Failures. Software, IEEE 25(5), 84-90 (2008)
18. Wieringa, R.: Design Science as Nested Problem Solving. In : Proc. DESRIST, NY (2009)
19. Hevner, A., Chatterjee, S.: Design Research in Information Systems. Theory and Practice. Springer, NY, USA (2010)
20. Sommerville, I.: Software Engineering 9th edn. Addison-Wesley, Boston, Mass. (2011)