An Experiment Design for Validating a Test Case Generation Strategy from Requirements Models

Maria Fernanda Granda
Department of Computer Science
University of Cuenca
Cuenca, Ecuador
fernanda.granda@ucuenca.edu.ec
Computing and Information Systems Department
Universitat Politècnica de València
Valencia, Spain
mgranda@dsic.upv.es

***Abstract*—Currently, in a Model-Driven Engineering environment, it is a difficult and challenging task to fully automate model-driven testing because this demands complete and unambiguous models as input. Although some approaches have been developed to generate test cases from models, they require rigorous assessment of the completeness of the derivation rules. This paper proposes the plan and design of a controlled experiment that analyses a test case generation strategy for the purpose of evaluating its completeness from the viewpoint of those testers who will use a Communication Analysis-based requirements model. We will compare the abstract test cases obtained by applying (i) manual derivation without derivation rules with (ii) manual derivation with transformation rules; and both these strategies against a case of automated generation using transformation rules.**

***Index Terms*—Experimental Design, Test Case Validation, Model-driven testing, Conceptual Schema Testing, Test Model Generation, Test Cases Generation.**

I. INTRODUCTION

Currently, in a Model-Driven Engineering environment, it is a difficult and challenging task to totally automatize model-driven testing [1] [2] because this demands complete and unambiguous models as input. Therefore, in practice, test cases are still derived manually from requirements specifications. However, this type of manual test case derivation is error-prone and very time consuming.

Our proposal for testing-based validation of conceptual schemas [3], takes advantage of the requirements models for generating test cases by using the Communication Analysis (CA) [4] method for modelling functional requirements.

A model transformation strategy was defined (see Fig. 1) to derive from CA requirements models initial versions of Test Models that can already be crossed to automatically generate abstract test cases [5]. Twelve transformation rules were defined to facilitate the generation of the test models and eleven refinement rules were defined for obtaining the abstract test cases from the test model (see Fig. 3).

In order to assess the completeness[1] of the transformation and refinement rules and, the algorithm to generate the test model, we propose an experimental design that involves different generation strategies from a requirements model.

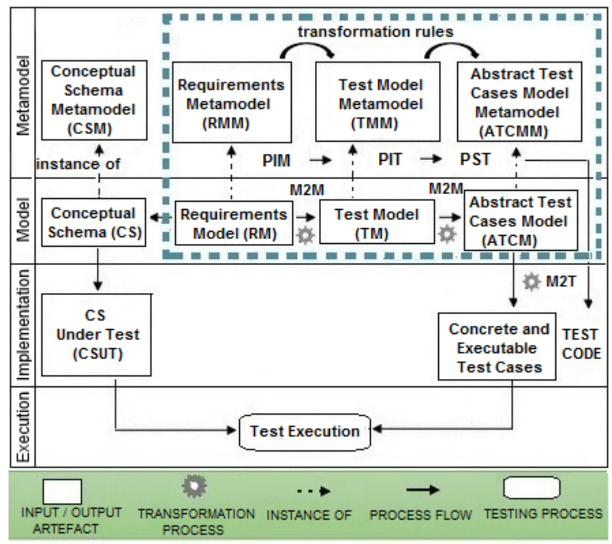

Fig. 1. An overview of our MDT approach from requirements models [5]

Based on the results obtained, we hope that future work will allow us: (i) to adjust the definition of the proposed transformation and refinement rules; and, (ii) to identify the strengths and limitations of the proposed testing approach.

Other proposals to measure the completeness of requirements exist in the literature [6] [7]. However, these works either use measurement strategies of the completeness of the requirements with respect to (i) the available input documents for the requirements definition process or (ii) the functional completeness of the future implementation of the system. In our case the completeness concerns to the transformation and derivation rules and algorithm used in the test case generation strategy for testing conceptual schemas.

The paper is structured as follows. Section II describes the experimental plan, and Section III summarizes the issues with the potential to threaten the validity of the experiment.

[1] Semantic Completeness means that it contains all the statements about the domain that are correct and relevant [11].

II. Experimental Plan

A. Experimental Goal

The goal of our empirical study, according to the Goal/Question/Metric Paradigm [8], is the following:

To analyse the generation strategy of the abstract test cases for the purpose of evaluating them with respect to their completeness from the testers's viewpoint.

B. Context Selection

This experiment will be conducted in the context of the Vrije University Amsterdam (Netherlands) and Universitat Politècnica de València (Spain) -researchers and testers (practitioners) willing to derive abstract test cases by using our model-driven generation strategy [5].

C. Subjects

We have planned to gather a minimum of 20 subjects for the empirical evaluation. As prior knowledge and experience in requirements engineering and testing are required, PhD students, senior researchers and testers are very welcome.

D. Objects

We have selected two requirement models of similar complexity. One model will be used for the CA training phase and another for the experimental phase.

E. Research Questions

By means of the study, we aim to be able to respond to the following general question:

- *RQ1:* How much of the functionality specified in the requirements model is covered by the generated abstract test cases?

As our research focus on the generation strategy, the following research questions are derived:

- *RQ1.1:* is there a significant difference between the degree of completeness of the abstract test cases generated manually with derivation rules and those generated manually without derivation rules?
- *RQ1.2:* is there a significant difference between the degree of completeness of the abstract test cases generated manually with derivation rules and those generated automatically?

F. Variables

We have identified four types of variables [9].

1) Response Variables

In order to answer these research questions, we will consider one response variable (dependent variable), which will be particularly related to the effectiveness of the semantic completeness of the transformation and refinement rules; and, the algorithm to generate the test model. This variable is defined as follows:

Semantic Completeness (*degCompl*): We measure this variable by counting how many functions expressed in the input specification are also treated in the abstract test cases.

To do this the reviewer will use a valid and correct requirements specification for the selected system.

2) Factors

We want to assess the impact of the following factor (independent variable) in our study.

Test cases derivation strategy: three different strategies for the derivation of test cases from requirement specifications will be used: manual derivation (MD) without derivation rules, manual derivation with transformation rules (MDR) and automated generation with transformation rules (AG).

3) Blocking Variables

*K*nowledge *level* of Communication Analysis. We expect all the subjects to start the training without any prior experience of CA and will assess their competence to block them. This variable will be measured by combining three different measures: (i) several CA exercises (ii) the result of a CA knowledge test.

4) Parameters

Similar Complexity and different size for selected requirements models.

G. Hypotheses

The hypotheses formulated from the research questions defined above are the following (suffixes are added to the variable name to indicate the technique: AG stands for automated generation, MD stands for manual derivation and MDR stands for manual derivation with derivation rules). Null hypotheses (Hi0) and alternative hypotheses (Hi1) are provided.

- $H_{1,0}$: There is no difference in the completeness of test cases generated manually with the derivation rules and those generated manually without derivation rules.
  $H_{1,0}$*: degCompl_MDR = degCompl_MD*
- $H_{1,1}$: Test Cases generated manually with derivation rules are more complete than manually without derivation rules.
  $H_{1,1}$: *degCompl_MDR ≠ degCompl_MD*
- $H_{2,0}$: There is no difference in the completeness of test cases generated manually with derivation rules and those generated automatically.
  $H_{2,0}$*: degCompl_MDR = degCompl_AG*
- $H_{2,1}$: Test Cases generated automatically are more complete than manually with derivation rules.
  $H_{2,1}$: *degCompl_MDR ≠ degCompl_AG*

H. Experimental Design

Having established the response variables, blocking variables and factors, the next step is to define the experimental design. This experiment is a between-subjects design (each participant is tested under one condition only) [10] and depends on one factor with three alternatives, each being a different technique for deriving test cases from a Communication Analysis requirements model.

1) *Manual Derivation (MD):* This group will take the requirements model and by means of a careful reading and primitive's analysis will facilitate the identification of business objects (classes of objects), operations (services and triggers), relationships of objects (links), constraints, etc. The abstract test cases will be derived in this way.
2) *Manual Derivation by Applying Transformation Rules (MDR):* This group will apply the set of defined

transformation and refinement rules, a test model with abstract test cases will be derived manually.

3) *Automated Generation by using the Tool Support (AG):* This group will apply the tool support, a test model with abstract test cases will be derived automatically.

I. Experimental Procedure

The experimental procedure is depicted in Fig. 2 and is explained as follows:

1) Session 1: Initial Session

We plan to start with an initial session, which will follow the same agenda with all the participants.

First, there will be a presentation in which general information will be given on the experiment and instructions will be issued on how to carry out the task.

The subjects will then proceed to fill out a survey with the purpose of identifying their background and experience in using requirement specification techniques. They will then be trained in Communicational Analysis, covering the concepts and modelling primitives of the Communicative Event Diagram and the Event Specification Templates, which are the main artefacts for abstract test case derivation. This task will be done by using one of the selected systems. At the end of the session an assessment of their training will be carried out. Only subjects who obtain a value equal to or greater than 7 will be considered for the next session.

2) Session 2: Experimental Session

In this session the subjects will be randomly split into two treatment groups: MD (manual derivation) and MDR (manual derivation with transformations rules) groups. The MDR group will be trained in manual derivation by using the transformation rules and refinement rules to derive the test model and abstract test cases. Their training will then be assessed.

The first part of the experiment will then be run. For this purpose, the subjects will proceed to generate abstract test cases manually with and without transformation and refinement rules.

In the second part of the experiment, automated generation of the test cases will be executed on the same model as that used in the manual derivation.

We will then compare all the abstract test cases generated by the different strategies (MD with AG and MDR with AG) in order to obtain the information on the differences and similarities with the test cases generated by experts in testing.

The subjects will then be answered to complete questionnaires in order to feedback on the derivation technique.

Finally, these artefacts will be used in the discussion, which aims to reach a consensus among the reviewers, and the response variable (completeness) will be measured.

J. Instrumentation

The following instruments will be used:

1) Surveys and Questionnaires

Web-based surveys and questionnaires will be provided to subjects by means of a link and using web forms.

2) Glossary of Primitives

This glossary is related to the primitives of the Communicative Event diagram and the Event Specification Templates, and also will be provided to the subjects to help them in the derivation process.

3) Guideline of the Transformation Rules

A guideline to illustrate the use of the transformation rules will be provided to the subjects (testers).

4) System Descriptions using Communicational Analysis

Two Communication Analysis requirements models will be provided in this experiment. One of the models will be used in the training exercises (initial session) and the other will be used in the second session.

With respect to the infrastructure needed, the following requirements should be covered: i) a computer (laptop or tablet) with WIFI access per subject. ii) Internet connection, iii) one video projector.

II. Threat Analysis

This section discusses possible issues that threaten the validity of our experiment [10].

A. Conclusion Validity

Conclusion validity is concerned with issues that affect the ability to draw the correct conclusions about relations between the treatment and the outcome of the experiment.

- Reliability of Measures could be affected due to poor instrument design. To detect this threat, we will conduct a reliability analysis on the instruments with a pilot study performed prior to the experiment, within the Research Center on Software Production Methods (PROS) of the Universitat Politècnica de València.

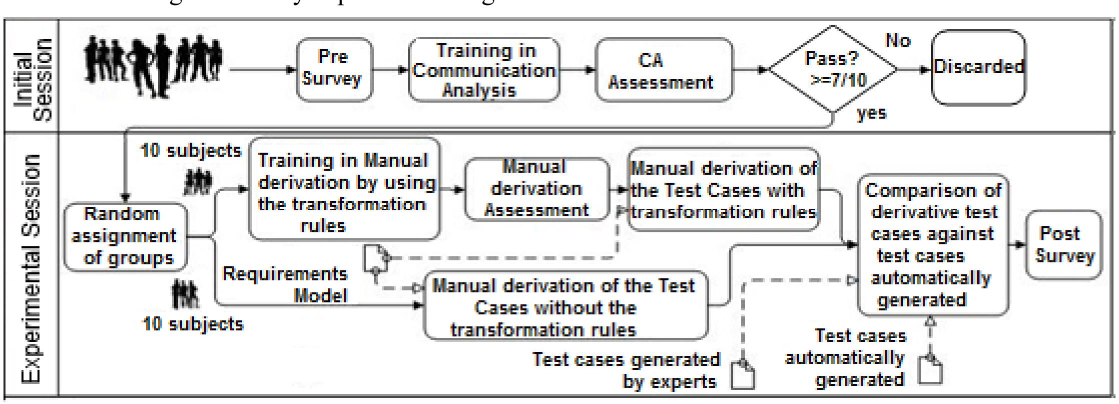

Fig. 2. Experimental Procedure

- There is a threat due to the heterogeneity of subjects, which could cause greater variability in measures (e.g. the participants' backgrounds). However, this heterogeneity would also contribute to the external validity of our study. We will verify that the subjects have a homogeneous background by means of a questionnaire. Moreover, as we have included a blocking factor in our design, this threat would be reduced partially.

B. *Internal Validity*

Internal validity is composed of influences that can affect the independent variable with respect to causality, without the researcher's knowledge.

- Instrumentation is the effect caused by the instruments used for the experiment. To avoid this effect, we will use web-based forms, which minimise transcription errors (the results can be directly downloaded).
- Selection is a risk related to the allocation of subjects to groups. We will apply a random allocation procedure with the intention of avoiding bias.

C. *Construction Validity*

Construct validity refers to the extent to which the experiment setting actually reflects the construct under study.

- As completeness can be measured by more one way to count the functions expressed in the input specification. It can become a threat if the abstraction level of the functions is not appropriately dealt. To avoid this threat, we will try to define metrics by considering several coverage types, such as all-paths and all-communicative events.
- Interaction of different treatments is the threat of the subjects being involved in more than one treatment. To avoid this, each group will apply only one treatment.

D. *External Validity*

External validity is related to the ability to generalise the results of the experiments.

- Interaction of selection and treatment is the perception of having a subject population not representative of the population we want to generalize to. In our experiment we plan to gather a minimum of 20 experienced subjects, which are representative tool´s users.
- Interaction of setting and treatment is the perception of the participants of the quality of the requirement models being influenced by any subjective issue. To avoid this threat, we will try to select representative requirements models.

ACKNOWLEDGMENTS

This work was sponsored by The Secretary of Higher Education, Science and Technology (SENESCYT: Secretaría Nacional de Educación Superior, Ciencia y Tecnología) of the Republic of Ecuador.

In addition, the author is indebted to Nelly Condori-Fernandez for her constructive comments.

REFERENCES

[1] Z. Dai, «Model-driven testing with UML 2.0,» de *Computer Science at Kent*, 2004.

[2] M. Mussa, S. Ouchani, W. A. Sammane y A. Hamou-Lhadj, «A Survey of Model-Driven Testing Techniques,» de *QSIC*, Jeju, Korea, , 2009.

[3] M. Granda, «Testing-Based Conceptual Schema Validation in a Model-Driven Environment,» de *CAiSE*, Valencia, 2013.

[4] S. España, A. González y O. Pastor, «Communication Analysis: a Requirements Engineering method for information systems,» de *21st International Conference on Advanced Information Systems (CAiSE'09)*, Amsterdam, The Netherlands, 2009.

[5] M. Granda, N. Condori-Fernandez y T. P. O. Vos, «Towards an Automated Generation of Abstract Test Cases from Requirements Models,» de *Workshop RET in 22nd IEEE International Requirements Engineering Conference*, Karlskrona, Sweden, 2014.

[6] S. España, N. Condori-Fernandez, A. González y O. Pastor, «Evaluating the Completeness and Granularity of Functional Requirements,» de *17th IEEE International Requirements Engineering Conference*, 2009.

[7] A. Ferrari, F. dell'Orletta y G. O. Spagnolo, «Measuring and Improving the Completeness of Natural Language Requirements,» de *REFSQ 2014*, 2014.

[8] V. Basili y H. Rombach, «The TAME Project: Towards Improvement-Oriented Software Environments,» *IEEE Trans. Software Eng,* vol. 14, nº 6, pp. 758-773, 1988.

[9] N. Juristo y A. Moreno, Basics of software engineering experimentation., 2001.

[10] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell y A. Wesslén, Experimentation in software engineering, Springer, 2012.

[11] O. Lindland, G. Sindre y A. Sølvberg, «Understanding quality in Conceptual Modeling,» pp. 42-49, 1994.

APPENDIX I. TRANSFORMATION AND REFINEMENT RULES

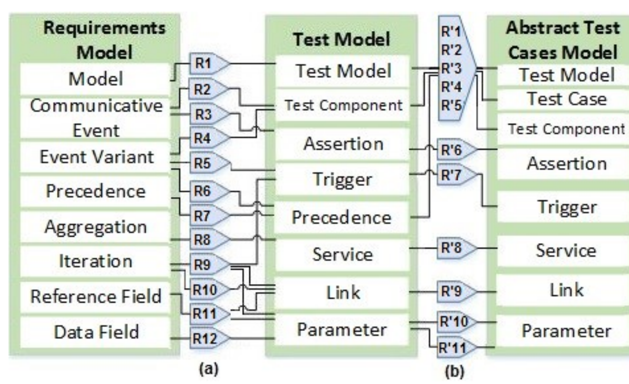

Fig. 3. a) Transformation rules and b) Refinement rules [5]