

C 5 - First International Workshop on Evolution Support for Model-Based Development and Testing (EMDT2010)

Time: Thursday, 16.09.2010

Location: Humboldt-Building, Lecture Room 202

Chairmen: M. Riebisch (D-Ilmenau)

9:00 a.m.	M. Riebisch, Qurat-ul-Ann Farooq, St.Bode, (D-Ilmenau)
Welcome and Introduction	
9:05 a.m.	B.-H. Schlingloff (D-Berlin)
Keynote: Model-Based Software Development – Perspectives and Challenges Model-based software development and testing has turned out to be the method of choice for safety-critical embedded systems. An abstract model reflects requirements and environmental conditions for the system. Such a model can be used in two ways - as a development model in a stepwise refinement process to derive the actual implementation, or as a testing model in order to derive test cases for some system under test. In this talk we discuss commonalities and differences between development models and testing models, discuss the formalization of requirements in models, and show how to automatically evaluate observations about a system with respect to a model. We illustrate our ideas with examples from aerospace, automation and medical devices. Finally, we discuss some recent trends and challenges in the area of model-based development and testing.	
9:50 a.m.	B. Güldali (D-Paderborn)
Invited Talk: Agility vs. Model-based Testing: A fair Play? “Agile manifesto” defines principles of a light-weight software development process for an improved customer satisfaction: adopting to changing requirements, frequently delivering working software in short cycles, and close collaboration of business people and developers. Automated testing plays an important role for assuring the quality of delivered software. Especially for the repeating activities of regression testing and integration testing, automated tools are indispensable for assuring short development cycles. “Model-based testing” is a widely accepted technique which extends the test execution of classical test automation by test design and test implementation. Thereby, logical and executable test cases are generated from formal test models. Thus, model-based testing can speed up the test automation in agile development processes. However, creating and maintaining the test models imply some challenges if model-based testing should be embedded into the agile development: How can we create test models efficiently? How can we assure the adoptability of test models? Can we use test models for communication with the business people? In this talk, we will discuss how model-based testing can support agile development without conflicting with the principles of agile manifesto.	
10:20 – 10:40 a.m. Coffee break	

10:40 a.m. | N. Kosindrdecha, J. Daengdej (T-Bangkok)

A Test Case Generation Technique and Process

It has been proven that the software testing phase is one of the most critical and important phases in the software development life cycle. In general, the software testing phase takes around 40-70% of the effort, time, and cost. This area is well researched over a long period of time. Unfortunately, while many researchers have found methods of reducing time and cost during the testing process, there are still a number of important related issues that need to be researched. This paper introduces a new test case generation process with a requirement prioritization method to resolve the following research problems: (a) inefficient test case generation techniques with limited resources (b) lack of an ability to identify critical domain requirements in the test case generation process and (c) ignore a number of generated test cases. In brief, the contributions are to: (a) study a comprehensive set of test case generation techniques since 1990 (b) concentrate on the model-based generation technique (c) define a new process to generate test cases (d) provide a guide for the direction of future research.

11:00 a.m. | Chr. Kop, G. Fliedl and H. C. Mayr (A-Klagenfurt)

From Natural Language Requirements to a Conceptual Model

In literature it is described in great detail how class diagrams and ER diagrams or UML class diagrams are derived from natural language sentences. It is normally assumed, that there is a direct correspondence between natural language elements (e.g. words) and conceptual model elements. We do not strictly follow this assumption because of the complexity of natural language with its ambiguities and ellipsis. Hence in this paper a stepwise generation of a conceptual schema out of natural language requirements sentences is proposed. According to the ideas of MDA we assume that automatic transformation steps from the source model (in our case natural language) to the target conceptual model (e.g., UML class diagram) make sense. In addition to that we suggest that the designer should play an important part during transformation. It is furthermore proposed to introduce an interlingua between the conceptual natural language and the UML class diagram.

11:20 a.m. | S. Roongruangsuwan, J. Daengdej (T-Bangkok)

Test Case Reduction Methods by Using CBR

Statistics gathered in past research show that testing, analysis, and debugging costs usually consume over 50% of the costs associated with the development of large software systems [5]. Many techniques have been proposed that reduce time and cost, such as test prioritization, regression selection and test reduction methods. Although, there are many proposed techniques, there are outstanding research issues that motivated this study: (a) redundancy test cases are still remaining (b) a size of test cases is still too big and (c) existing reduction methods consume a great deal of time. This paper introduces a new process, called "3R-2S-3R-2R". Also, it proposes reduction methods using case-based reasoning. In brief, the contributions are to: (a) study a comprehensive set of existing reduction methods (b) introduce a new concept of CBR for removing non-necessary test cases (c) define a new reduction process (d) propose and compare reduction methods and (e) guide future works.

11:40 a.m.	M. Riebisch, Qurat-ul-Ann Farooq, St. Bode (D-Ilmenau)
Introduction to the goals of the group discussion	
12:00 noon - 1:30 p.m. Lunch	
1:30 p.m.	All participants
Group discussion part I	
2:50 - 3:50 p.m. Coffee break	
3:50 p.m.	All participants
Group discussion part II	
4:30 p.m.	All participants
Conclusion of the discussion, deliberation of further steps	
5:00 p.m. End of Lecture Session	