International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:2, 2008

Suitability of Requirements Abstraction Model (RAM) Requirements for High-Level System Testing

Naeem Muhammad, Yves Vandewoude, Yolande Berbers, and Robert Feldt

***Abstract*—**The Requirements Abstraction Model (RAM) helps in managing abstraction in requirements by organizing them at four levels (product, feature, function and component). The RAM is adaptable and can be tailored to meet the needs of the various organizations. Because software requirements are an important source of information for developing high-level tests, organizations willing to adopt the RAM model need to know the suitability of the RAM requirements for developing high-level tests. To investigate this suitability, test cases from twenty randomly selected requirements were developed, analyzed and graded. Requirements were selected from the requirements document of a Course Management System, a web based software system that supports teachers and students in performing course related tasks. This paper describes the results of the requirements document analysis. The results show that requirements at lower levels in the RAM are suitable for developing executable tests whereas it is hard to develop from requirements at higher levels.

***Keywords*—**Market-Driven Requirements Engineering, Requirements Abstraction Model, Requirements Abstraction, System Testing.

I. Introduction

IN traditional development methodologies, testing is considered to be the last phase. In such methodologies, testers design or at least execute tests in the last phase. Recent approaches to software testing show that testing activities can be started in parallel to the requirements engineering activities. The V-Model [1] maps testing activities with the software development phases of the waterfall model. This model suggests that acceptance tests can be designed as soon as requirements are available. Starting testing activities at an early stage provides many benefits, such as finding and fixing bugs in requirements before system implementation [2]. Fixing bugs before system implementation is cheaper [2], since addressing them at later stages can result in major changes in the design of the system [3]. Moreover, developing tests in the last stage may increase pressure on the testing team, which may affect their efficiency [4]. It is therefore very cost effective to use requirements for designing tests at a very early stage in development [5,10].

Market-Driven Requirements Engineering (MDRE) is concerned with the requirements engineering of products targeted for the mass market. The requirements for such products are elicited from various stakeholders (e.g. users, developers and marketers), which often operate on different levels of abstraction [6]. Managing such requirements with different levels of abstraction is a difficult task [7], MDRE further complicates things because new requirements are continuously arriving in the requirements engineering process [8].

The Requirements Abstraction Model (RAM) [6,7,9] developed by Gorschek and Wholin is a model for MDRE that helps in managing different levels of abstraction among the requirements and that can handle the continuous arrival of new requirements. RAM provides four levels of abstraction on which the requirements can be placed: product level, feature level, function level and component level.

In view of the benefits of early test development, it is important to analyze to what extent RAM requirements can be used to design efficient tests. A study on understanding the suitability of RAM requirements for testing will be of great value in generalizing this model.

This paper investigates the suitability of RAM requirements for designing high-level tests. To do so, we examined a RAM requirements document of a Course Management System (CMS). The CMS is a web-based software system that supports teachers and students in performing course related functions (e.g. add a course, add a course participant and view course calendar) [11]. The RAM model has already been implemented in some companies that have requirements documents according to the RAM specification. However, CMS is the only RAM requirements document that was accessible to us.

Major tasks of the suitability investigation include selection of a test case template, selection of sample requirements, test case development and test case grading based on a chosen criterion. Test cases were graded on the basis of the level of

Naeem Muhammad is with Departement Computerwetenschappen, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium (e-mail: Naeem.Muhammad@cs.kuleuven.be).

Yves Vandewoude is with Departement Computerwetenschappen, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium (e-mail: Yves.Vandewoude@cs.kuleuven.be).

Yolande.Berbers is with Departement Computerwetenschappen, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium (e-mail: Yolande.Berbers@cs.kuleuven.be).

Robert Feldt is with School of Engineering, Blekinge Institute of Technology, SE- 372 25 Ronneby, Sweden (e-mail: Robert.Feldt@bth.se).

detail (extracted from the given requirement) they hold. The results indicate that most of the time it is possible to design executable test cases from the RAM requirements.

The remainder of the paper is organized as follows. Section II describes the Requirements Abstraction Model (RAM). Section III describes the requirements document analysis of a RAM requirements document and it elaborates the different tasks of the requirements document analysis. Section IV describes the related work. Section V concludes the study and outlines the future work.

II. Requirements Abstraction Model (RAM)

In Market Driven Requirements Engineering (MDRE), the requirements are expected to come from various sources and they have various levels of abstraction. This makes it hard to manage the requirements. Gorschek and Wohlin have developed the Requirements Abstraction Model (RAM) to manage requirements abstraction and the continuous arrival of requirements in the MDRE environments. The need for this model originated from the problems faced at Danaher Motion Särö AB (DHR) though the model is flexible and can be tailored for other organizations.

A. Structure of the RAM

MDRE development revolves around products in which different versions (releases) are produced over time. Potential stakeholders in a MDRE process may include end users, marketing teams, product managers etc. The requirements elicited from these sources may vary in their level of abstraction, as they are from different domains of knowledge. RAM provides four different levels of abstraction on which the requirements are placed:

1) Product Level

Requirements that are highly abstract (i.e. they represent product goals) are placed at the Product Level. The high abstraction in the product level requirements makes them directly comparable with product strategies and indirectly comparable with organizational strategies. This comparison can help managers to include or exclude any requirement and, moreover, to set the priorities of the requirements. A typical example of a product level requirement from the CMS is: "The product (software system) shall provide information to interested authorized system users about a course".

2) Feature Level

Requirements which are features of the product are placed at the feature level. Features are characteristics of the system; requirements at this level provide abstract descriptions of these characteristics. "It shall be possible to attach news items to a course" is an example of the feature level requirement.

3) Function Level

The third level in the RAM contains functional aspects of the requirement. At this level each requirement is described in such a way that it clearly shows what a user or system can do. Description at this level can be used to develop a design of the system. Moreover, requirements at this level are supposed to be testable. "Only system administrators or the course administrator shall be able to add or remove participants to or from a specific course" is a function level requirement of the CMS.

4) Component Level

The last level in the RAM model is the Component Level. Requirements at this level have detailed information. Component level requirements generally describe how a problem is to be solved. An example of the component level requirement from the CMS system is "If the user enters an incorrect user id and /or password the login page shall be reloaded with information showing that an incorrect login has been attempted".

Fig. 1 Requirements Abstraction Model (RAM) Abstraction Levels [7]

B. RAM Action Steps

Using RAM in the requirements engineering process involves three action steps, as shown in Fig. 2, execution of these action steps results in a requirement being placed at a particular level and linked with other requirements at adjacent level(s). The following is a brief description of these action steps.

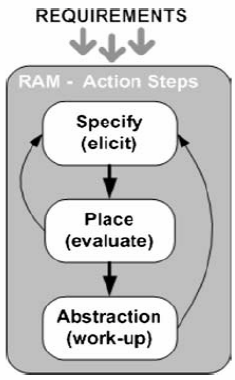

Fig. 2 RAM Action Steps [7]

1) Specify (Elicitation)

The first action step is to specify the raw requirements in a uniform way. To get a basic overview of the requirements, four attributes (more attributes are added in later stages) are specified in this action step. These attributes include Title, Description, Reason/Benefit/Rationale and Restrictions/Risks. The last two attributes are optional.

2) Place (Evaluation)

Place (evaluate) is the second action step of the RAM model. Each requirement is analyzed against a set of questions for each abstraction level starting at the product level. As soon as one question is answered positively, the requirement is placed at the corresponding level and the analysis is stopped. Some of the questions that are asked for product, feature, function and component levels are:

- **Product Level**

Is the requirement abstract enough to be comparable to the product strategies?

- **Feature Level**

Does the requirement describe what the system should include?

- **Function Level**

Does the requirement describe the functionality that is to be performed by the user?

- **Component Level**

Does the requirement consist of a specific suggestion as to how something should be done/solved?

3) Abstraction (Work-Up)

The third action step of the RAM is called abstraction, which is also known as work-up activities. In this action step, the requirements are abstracted or broken-down on the basis of their placement. New requirements called work-up requirements are created during these activities. Subsequently, a requirement is linked with existing or newly created requirements at adjacent level(s). In addition, to make requirements more detailed, additional attributes (Requirement Source, Due Date and Dependency) are added at this stage.

An important goal of the RAM is to make each requirement comparable with product strategies. To do so, RAM enforces the following two rules:

R1. "No requirement may exist without having a connection to the Product Level."
R2. "All requirements must be broken down to the Function Level."

To comply with R1, each requirement at a lower level is abstracted upward and linked with existing or work-up requirements at adjacent levels of the RAM. To fulfil rule R2, requirements at higher levels in the RAM are broken down and linked with other requirements at adjacent levels. The abstraction action step results in requirements that have explicit or implicit links (through other requirements) with the product strategies. Since requirements at this stage exist in chains, the removal of any requirement or addition of a new requirement requires action steps to be executed again.

III. Suitability of RAM Requirements for High-Level Testing

To understand the suitability of RAM requirements for high-level testing, a requirements document analysis of a CMS was performed. The following sections describe the activities of our requirements document analysis and its application on the CMS.

A. Requirements Document Analysis Method

The requirements document analysis method used for this study consists of 4 steps: selection of sample requirements, selection of a test case template, selection of a test case grading criterion, and test case designing process. The following sections will describe each of these steps.

1) Selection of Requirements

Selection of requirements is the first activity of the requirements document analysis. For this study, sample requirements have been selected from a Course Management System requirements document. We have randomly selected 5 requirements from each RAM level (20 requirements in total). Due to the size limitation of the paper, we provide only one requirement from each level here (see Table I).

TABLE I
Sample RAM Requirements, Taken From [11]

ID	Level	Title	Description	Rationale	Restriction
1	Product	Swedish Market	The product is targeted towards the Swedish market		This may change in coming releases of the system.
2	Feature	Product Access	Only users with the right privileges shall be able to view, add, edit or remove content in the product.		
3	Function	Access to add and remove course participators	Only system administrators or the course administrator shall be able to add or remove participants to or from a specific course.		
4	Component	Successful Login	When a user has successfully logged in, the product shall display the user's personal start page.	After user authentication, the user wants to start working.	

2) Selection of the Test Case Template

The template for specifying a test case varies from

organization to organization. It is hard to incorporate every element of various test case templates used in different organizations. Thus we document our test cases using a test case template which is based on the IEEE 829-1998 [12] test specification standard. As the standard template is mostly theoretical, we modified it to describe the actual test execution. We replaced Output specifications, Environmental needs, Special procedural requirements and Intercase dependencies elements of the standard specification with Test case purpose, Steps and Expected results. This replacement was required in order to describe the reasons for the design of the test and how the test case is to be executed. Table II contains the modified test case template.

TABLE II
TEST CASE TEMPLATE

Test Case ID	A unique identifier of the test case.
Test Case Purpose	A brief description of the purpose of the test case.
Customer Requirement	Brief description of the items and features of the system selected to be tested through this test case.
Preconditions	Assumption or conditions required to be met before the execution of the test.
Inputs	List of input data required to execute the test case, which may include variable values, files, etc.
Steps	Listing of the steps required to carry out the test.
Expected Results	Results expected from the execution of the test case.

3) Test Case Grading Criteria

The test cases will be analyzed for the level of detail they contain and then graded on a scale of 5 to1. The scale of 5 to 1 grading is as follows:

5: The test case contains detailed information for each of its fields. The test case has clear preconditions, correct input data, obvious steps through which the test case runs and comprehensible expected outcomes. Moreover, all of the information has been extracted directly from the given requirement.

4: The test case has information for some fields and it is in executable form. However, some information is missing which is not directly available from the given requirement. This missing information can easily be gathered from other requirements or from the context of the requirement.

3: The test case lacks information and is not in executable form. However, missing information can be inferred from other requirements or from the context of the requirement. Adding this information can make the test case executable.

2: The test case has some information extracted directly from the given description of the requirement but it cannot be considered as executable. Moreover, it is not possible to extract more information for this test case from other requirements in the requirements document.

1: The test case is very incomplete and does not provide clear understanding of how to test the requirement. In certain cases, information given with the requirements does not provide an understanding of what to test and how to test. It is quite difficult to design test cases from such requirements. Test cases designed from such requirements are very ambiguous and generally state only the test purpose (goal).

4) Requirements Document Analysis Process

For each test case, the fields of the test case template are filled in by manually extracting information from the corresponding requirement. If insufficient information is present for making a detailed test case, other requirements in the document are used as additional sources of information.

It is important to note that information for each test case must be extracted only from the given requirement, from the context of the requirement and from the other requirements in the requirements document. The author is not allowed to add information from his/her own experience. In this way, the test cases will be designed from each selected requirement. Subsequently, the test cases will be analyzed against the grading criteria and a suitable grade will be determined for each test case.

B. Application of the Requirements Document Analysis Method

We have applied the method described in section 3.1 to develop test cases from the sample requirements. Although multiple test cases can be designed from a single requirement, this study has been restricted to one test case per sample requirement. Thus, a total of 20 test cases have been designed from the 20 sample requirements. Due to the size limitation of the paper, it is not possible to discuss all 20 test cases here. We will develop, grade and discuss test cases from one requirement at each level.

Note that in the following test cases, information extracted from other requirements is put in italics.

1) Test Case from Product Level Requirement

Table III contains a test case designed from the product level requirement “Swedish Market”. The highly abstract nature of the test case only allows for the extraction of the test case purpose. Moreover, other requirements in the document do not provide any support to fill the remaining fields of the test case. With very incomplete information, the test case is assigned grade 1.

TABLE III
TEST CASE FROM PRODUCT LEVEL REQUIREMENT

Test Case ID	TC.1
Test Case Purpose	Test that the system supports the Swedish language
Customer Requirement	1
Preconditions	-
Inputs	-
Steps	-
Expected Results	-

2) Test Case from Feature Level Requirement

This test case has been designed from the "Product Access" feature level requirement. Along with the test case purpose, we can infer some information for preconditions and inputs from the description of the requirement. The description suggests that a user should be registered (precondition). In addition, viewing, adding and removing course content implies that a course ID (input) is required for all these tasks. Although the test case has information for the test case purpose, precondition and input, it lacks important information such as the steps required to execute the test. However, the use of other requirements in the document can help in extracting the missing information. For instance, we can identify possible expected results from the "Course start page content" requirement. This requirement stipulates that the start webpage of a course should contain links to many tasks which a user can perform (e.g. a links to course news, file archive, course calendar and discussion forum). In addition, "Personal start page content" and "Successful login" requirements provide actions required to view the course content (necessary steps to execute the test case). These requirements describe how a user can login to the system and what content a personal start page contains.

Adding information from other requirements results in a detailed (executable) test case. Therefore, it is assigned grade 3.

TABLE IV
TEST CASE FROM FEATURE LEVEL REQUIREMENT

Test Case ID	TC.2
Test Case Purpose	Test that a course participator (user) can view course content.
Customer Requirement	2
Preconditions	User is registered
Inputs	Course (ID or name), *User ID*, *User password*
Steps	*1. Enter user ID* *2. Enter password* *3. Click on login button* *4. Select the course with the given course ID or name*
Expected Results	*The user will see a web page containing course news, file archive, course participators, course calendar, course information, course literature list, course links and discussion forum.*

3) Test Case from Function Level Requirement

The test case given in Table V has been designed from the function level requirement "Access to add and remove course participators". Test case purpose, preconditions and inputs can easily be identified from the description of the requirement. However, the test case is incomplete because it lacks information for some fields. Exploring other requirements in the document can provide the missing information. Steps required to execute the test can be inferred from the "Personal start page", the "Course start page" and the "Manage course participators" requirements. The first two requirements describe what content a personal and course start webpage contains.

The "Manage course participators" requirement describes that a system administrator and a course administrator can add a new participant to a course. With these requirements it is possible to determine the sequence of events all the way from the login of an administrator until the addition of a participant. Since the support of other requirements was required to design a detailed (executable) test case, this test case is assigned grade 3.

TABLE V
TEST CASE FROM FUNCTION LEVEL REQUIREMENT

Test Case ID	TC3
Test Case Purpose	Test that a course administrator can add course participators to a course.
Customer Requirement	3
Preconditions	Course administrator exists, Course exists, *User is at personal page*
Inputs	Course (ID or name), User (to add as participator)
Steps	*1. Open course start page* *2. Click on course participator link* *3. Add course participator*
Expected Results	-

4) Test Case from Component Level Requirement

Table VI contains a test case designed from the "Successful login" component level requirement. The description of the requirement is clear enough, and it is possible to infer information for all fields of the test case. Since this test case contains detailed information (extracted directly from the given requirement), the test case can be assigned grade 5 on the scale.

TABLE VI
TEST CASE FROM COMPONENT LEVEL REQUIREMENT

Test Case ID	TC4
Test Case Purpose	Test that when a registered user provides correct user ID and password, then he/she views the personal start page.
Customer Requirement	4
Preconditions	User exists, User is not logged in
Inputs	User ID=valid user , User Password=valid password
Steps	1. Provide User ID 2. Provide User Password 3. Login
Expected Results	User sees personal start page

C. Test Cases Grade

Table VII contains the grades of all 20 test cases. We will discuss the results for each level individually, moving from component level to product level.

TABLE VII
TEST CASES GRADES

RAM Level	Grade 5	Grade 4	Grade 3	Grade 2	Grade 1
PRODUCT	-	-	-	2	3
FEATURE	-	-	3	2	-
FUNCTION	-	1	4	-	-
COMPONENT	4	-	1	-	-

80% of the component level requirements (independently) allowed for the design of detailed test cases. This shows that component level requirements hold information which is suitable for designing test cases. Information provided for component level requirements clearly describes what to test and how to test.

For function level requirements, most of the time (80%), support of other requirements was required to design detailed tests. Generally, these requirements provide a clear understanding of what is to be tested, the possible inputs and the preconditions for the test. However, the contribution of other requirements is required to fill in information for remaining fields.

Three test cases designed from feature level requirements have grade 3, whereas 2 test cases have grade 2. None of the feature level requirements could help in designing executable test cases individually. However, for three test cases, support from other requirements made it possible to convert incomplete test cases into complete. The results show that feature level requirements only describe what is to be tested, but not how to test.

Results show that product level requirements are least suitable for designing test cases. None of the selected requirements could produce an executable test case. Three product level requirements could only provide test case purpose. Moreover, it was not possible to complete these test cases with the help of other requirements in the document.

Our study highlights an important issue concerning the criteria used to place a requirement on one of the four abstraction levels of the RAM. It appears that test cases generated from the feature and function level requirements show similar testability (the majority of test cases from both levels have grade 3). This suggests that the questions used to distinguish between the function and the feature levels of RAM are not good enough. We therefore suggest a revision of these criteria to make the RAM model more effective.

In addition, RAM currently requires the breakdown of requirements down to the function level. Since component level requirements are better suited to generate test cases, it may be worthwhile to break requirements down to the component level instead.

IV. RELATED WORK

Although no explicit work had previously been performed to evaluate the RAM requirements for testing purposes, studies exist which evaluate other perspectives of the RAM.

Gorschek et al performed an industry evaluation of the RAM model [13]. The evaluation was made by using RAM in two separate requirements engineering processes, at DHR and ABB. Evaluation includes the study of the effects of using the RAM model in a requirements engineering process and the effects on quality of the requirements. The results of the study showed that RAM can improve a requirements engineering process. They also proved that RAM can be tailored for various organizations. Although they evaluated the RAM model from a different perspective, their results provided inputs to our study. Ultimately, the results of both studies are targeted towards generalizing the RAM model.

A large body of literature exists on test generation without using the RAM model. We describe two of these studies below.

Tahat et al [14] developed a method for automatic requirement-based black-box test generation. The method automatically converts individual requirements (textual and SDL formats) into SDL (Specification and Description Language) system models. In turn, SDL models are transformed into EFSM (Extended Finite State Machine) models. In the last step, test cases are automatically generated from EFSM models. The potential benefits of this approach are a reduction in the number of test cases and the better quality of the tests. The proposed method automatically generates test cases from requirements in both natural and formal languages, whereas we designed test cases manually from requirements in natural language only. Since the results of their method are very promising, it seems feasible to replace our manual test designing with this method for future requirements document analysis of RAM requirements documents.

Ryser and Glinz [15] proposed a method called SCENT, which systematically generates test cases from scenarios. The SCENT method first generates natural language scenarios from given requirements. Afterwards, these scenarios are formulized using state-charts, and test cases are generated by traversing paths in state-charts. The authors applied their SCENT method in two projects, and showed that their method can generate test cases from early phase artefacts (requirements document) in a systematic way. Since the SCENT method provides a systematic way of generating test cases from requirements, thus replacing our manual test design (that we used for designing test cases from RAM requirements) method with this can make our test designing process more efficient. Although the results of this method are very promising, some issues still need to be addressed. Generally, scenarios describe the functional behaviour of the system, and thus the SCENT method is not very effective for generating test cases from non-functional requirements.

V. Conclusion and Future Work

In this paper, we have analyzed the suitability of the RAM requirements for high level testing. For this purpose, we have chosen sample requirements from the requirements document of a course management system. We have designed test cases from the selected RAM requirements, and graded them based on the bases of the level of detail they contain.

The results of the study show that the testability of the requirements decreases as we move upward, from the component level to the product level. The component level requirements proved to be a good source for designing tests. In contrast, the product level requirements could not provide adequate support for designing tests. The function and feature level requirements showed same behaviour, i.e. individually, these requirements could not provide better support for test designing.

In total, 65% of the test cases were in executable form, either from the single requirement or the multiple requirements. These statistics illustrate that the RAM requirements are suitable for designing high-level tests. The results of our study will help in generalizing the RAM model, and they will provide input into the process of making the decision about its adoption.

In this paper, we have analyzed only a single requirements document (Course Management System). We have a plan to analyze more requirement documents from different domains in the near future. This will help in understanding the comparative behaviour of the results, (i.e. how the results of different requirement documents vary).

References

[1] Pyhajarvi, M., Rautiainen, K., Itkonen, J.: Increasing Understanding of the Modern Testing Perspective in Software Development Projects. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03), (2003).

[2] Offutt, A.J., Yiwei, Xiong, Shaoying, Liu: Criteria for Generating Specification-Based Tests. In: 5th International Conference on Engineering of Complex Computer Systems (ICECCS '99), pp. 119-129 (1999).

[3] Ryser, J., Berner, S., Glinz, M.: On the State of the Art in Requirements-based Validation and Test of Software. Technical Report, Institut fur Informatik, University of Zurich (1998).

[4] Elfriede, D.: Effective Software Testing: 50 Specific Ways to Improve Your Testing. Addison Wesley Professional, (2002).

[5] Tahat, L.H., Vaysburg, B., Korel, B., Bader, A.J.: Requirement-based automated black-box test generation. In: 25th Annual International Computer Software and Applications Conference (COMPSAC'01), pp. 489-495 (2001).

[6] Gorschek, T., Svahnberg, M., Borg, A., Loconsole, A., Borstler, J., Sandahl, K., Eriksson, M.: A Controlled Empirical Evaluation of A Requirements Abstraction Model. In: Information and Software Technology, Vol. 49, pp. 790-80 (2007).

[7] Gorschek, T., Wohlin, C.: Requirements Abstraction Model. In: Requirements Engineering, Vol. 11, pp. 79-101 (2006).

[8] Karlsson, L., Dahlstedt, A.G., Natt och Dag, J., Regnell, B., Persson, A.: Requirements Engineering Challenges in Market-Driven Software Development - An Interview Study with Practitioners. In: Information and Software Technology, Vol.49, pp. 588-604 (2007).

[9] Gorschek, T., Wohlin, C., Garre, P.: A Model for Technology Transfer in Practice. In: IEEE Software, Vol. 23, pp. 88-96 (2006).

[10] Abdurazik, A., Ammann, P., Ding, W., Offutt, J.: Evaluation of Three Specification-based Testing Criteria. In: Sixth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '00), pp. 179-187 (2000).

[11] Svahnberg, M.: Course Management System RAM Requirements Document, Blekinge Tekniska Hogskola Sweden.

[12] IEEE, "IEEE-STD 829-1998: IEEE standard for software test documentation," IEEE, [online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=5976. [Accessed: August 08, 2007].

[13] Gorschek, T., Garre, P., Larsson, S.B.M., Wohlin, C.: Industry Evaluation of the Requirements Abstraction Model. In: Requirements Engineering, Vol. 12, pp. 163-190 (2007).

[14] Tahat, L.H., Vaysburg, B., Korel, B., Bader, A.J.: Requirement-Based Automated Black-Box Test Generation. In: 25th Annual International Computer Software and Applications Conference (COMPSAC'01), pp. 489-495 (2001).

[15] Ryser; j., Glinz, M.: SCENT: A Method Employing Scenarios to Systematically Derive Test Cases for System Test. Universitt Zrich, Institut fr Informatik, Zrich, Berichte des Instituts fr Informatik, [online]. Available: http://citeseer.ist.psu.edu/ryser00scent.html. Accessed: November 24, 2007].