A technique for the generation of availability scenarios in the ATAM

Oswaldo Cabral de Vasconcellos Neto, Paulo Sérgio Muniz Silva

Escola Politécnica da Universidade de São Paulo – Depto. de Engenharia de Computação e Sistemas Digitais
{oswaldo.vasconcellos,paulo.muniz}@poli.usp.br

Abstract. Availability is an important non-functional requirement to be considered in the development of software systems that provide e-business services on the Internet. In ATAM, a method for software architecture evaluation, this requirement is analyzed by means of availability scenarios. This paper aims to establish an analysis technique for a class of Internet software systems that enable the identification of elements to generate availability scenarios. The technique is based on dependability concepts, structured by elements belonging to the NFR framework, with a hierarchical model approach to availability.

Keywords: Software architecture evaluation, non-functional requirements, dependability

1 Introduction

The use of the Internet for e-business service automation has been adopted as a strategy by organizations in several sectors of the economy. To attain maximum performance from this distribution channel, the availability of e-business services is extremely important. The systems must be capable of maximizing the time they are available, even in the presence of faults, by employing fault prevention and recovery techniques. For banking applications, for instance, the acceptable downtime per year is only 5 minutes, which corresponds to a 99.999% availability. [9].

Availability, a non-functional requirement, may be defined as readiness for correct service [1]. It is an attribute of dependability, which may be defined as the ability to deliver service that can justifiably be trusted [8]. Such a capacity is heavily dependent on system architecture and, especially, on software architecture, whose design must therefore take availability-related aspects into account.

One way of analyzing solutions for building computer system architectures is the evaluation of the architectural design with quality attribute scenarios [3]. One scenario is comprised of a stimulus representing an event or a condition that the architecture must respond to, and of a response that corresponds to the activity executed after the occurrence of a certain stimulus. For availability, [3] presents characteristics that may be mapped to availability scenario elements. The faults may be considered stimuli for scenarios and the responses are possible reactions to the

occurrence of these faults, which are associated with measures such as steady-state availability and repair time.

One method for architecture evaluation that is based on the employment of quality attribute scenarios is the ATAM (*Architectural Tradeoff Architectural Analysis*) [7]. Its main objective is to map the consequences of architectural decisions to the quality attributes, or non-functional requirements, of the system, thus identifying possible tradeoffs among different quality attributes. In the application of the method, the involvement of stakeholders during the generation of quality attribute scenarios is crucial. In [5], it is stated that the evaluation quality of a software architecture depends mostly on the "quality" of the stakeholders taking part in the evaluation. This evaluation is a complex task, as it deals with relationships between non-functional and functional requirements, synthesized in architectural decisions. In the event that there is no technique enabling an evaluation to be guided in a systematic fashion, any evaluation of the architecture exclusively based on the experience of the stakeholders may be compromised, since important scenarios concerning non-functional requirements may not be considered.

Trying to minimize the risks involved in dependence on stakeholders and to aid in the generation of scenarios involving the non-functional requirement availability in ATAM method, this paper proposes a technique that enables the generation of availability scenarios for a class of e-business software systems. This technique, entitled WSSAA (Web Software Systems Availability Analysis), helps in the generation of availability scenarios in a detailed and organized manner, thus enabling the identification and characterization of failures that will be employed as stimuli for availability scenarios.

The ATAM method is briefly introduced in section 2 and the proposed technique for the generation of availability scenarios is described in section 3. Section 4 provides final considerations and comments on a significant related work.

2 The ATAM Method

Quality attribute scenarios are employed in the ATAM method in a simplified manner, when compared to the original definition of scenarios proposed in [3]. In the ATAM, scenarios are composed of stimuli, environment and response. In [7] it is emphasized the importance of the employment of quality attribute scenarios in order to precisely elicit goals related to the quality attributes that will be evaluated when the method is applied. Due to space limitations, we present a brief overview of the ATAM steps [5]:

1. Present the ATAM method;
2. Present business drivers;
3. Present the architecture;
4. Identify the architectural approaches;
5. Generate the quality attribute utility tree - the generation of a utility tree to elicit scenarios enabling the characterization of system quality attributes. The utility tree

is a top-down mechanism employed to translate the business drivers of a system into concrete quality attribute scenarios;

6. Analyze the architectural approaches - based upon the high-priority scenarios identified in Step 5, the architectural approaches addressing those scenarios are elicited and analyzed. During this step, architectural risks, non-risks, sensitivity points and tradeoffs are identified for each scenario;
7. Brainstorm and prioritize scenarios - a larger set of scenarios is elicited and prioritized from the entire group of stakeholders. In this stage, new quality attribute scenarios may be identified, which may then be incorporated into the utility tree;
8. Analyze the architectural approaches - this step reiterates step 6, but considers the scenarios prioritized in step 7 as test cases for the analysis of the architectural approaches adopted;
9. Present the results.

The essence of the method is contained in steps 5, 6 and 7. In step 5, evaluators work together with the project decision makers (system architects, managers, client representatives etc.) in order to identify, prioritize and refine the quality attribute scenarios. The identified scenarios are prioritized and organized into a utility tree, which serves as a basis for the definition of requirements for quality attributes. In step 6, the architectural approaches identified in step 4 are applied to the scenarios obtained in step 5. In step 7, the scenarios are prioritized.

3 The Proposed Technique for the Generation of Scenarios

The WSSAA technique is intended to provide elements for the generation of availability scenarios for software architectures that support e-business services. The technique makes use of the dependability concepts defined in [1] and [8], structured and organized according to elements of the NFR framework described in [4]. [6] describes an architectural framework for modeling the availability of the class of e-business software systems considered in the proposal. The NFR framework provides a qualitative process to represent and analyze non-functional requirements based on the premise that such requirements are not always absolutely satisfied. In order to express such a premise, it represents non-functional requirements as *softgoals*, which may be refined, interconnected and analyzed on a graph called SIG (Softgoal Interdependency Graph). Essentially, the refinement reflects the knowledge, the reasoning process and the subsequent design decisions concerning the non-functional requirements. When refined, a derived softgoal may positively or negatively contribute to - or affect - the degree of satisfaction of a softgoal at a higher level in the refinement chain. Presuming that satisfaction is not absolute, but is within qualitatively interpreted acceptable limits, it can be said that a softgoal is *satisficed*, i.e. not absolutely satisfied. The analysis of satisficeability of softgoals is carried out with an evaluation procedure that semi-automatically determines the impact of design decisions on the fulfillment of the non-functional requirements [4].

In general terms, there are two aspects of interdependency in the NFR framework: refinements and contributions [4]. Decomposition refines a softgoal into other softgoals of the same kind (for example, accuracy). The kinds of decomposition correspond to the kinds of softgoals defined in the framework: non-functional requirements, operationalizing and argumentation. The contribution outlines how one softgoal satisfices a higher softgoal in the refinement chain. Several kinds of contributions are defined, such as the AND, OR, MAKE etc. contributions.

The use of elements from the NFR framework in the proposed technique enables the structuring of dependability concepts and characteristics present in the analyzed software architecture, in such a manner that the application of an evaluation procedure enables the determination of the impact of threats to dependability on non-functional requirements softgoals.

Another conceptual base of the WSSAA technique is a hierarchy with different levels for modeling availability of e-business systems, as described in [6]. This hierarchy has four levels - the user level, the function level, the service level and the resource level - and it is employed in the proposed technique in decomposition methods for availability softgoals.

In this article, the WSSAA technique is initially restricted to e-business software systems with a three-layered architecture. The first layer follows the logic of presentation and is directly related to the interaction between components responsible for interface generation between the system and its users, involving visualization and control aspects of Web pages. The second layer is directly related to the processing of the business rules inherent to the application domain, and involves the use of load balancers, Web servers and application servers. Finally, the integration layer concerns the communication of data, enabling integration with other systems or databases. The proposed technique consists of the following elements:

1. *Non-functional requirements softgoals (NFR softgoals).* The technique employs NFR softgoals, as defined in [4], for the representation of attributes of dependability, as defined in [8]. The representation of softgoals uses the same notation employed in the NFR framework, with the graphic representation of a cloud, and the syntactical description *Type[Topic]*. *Type* represents the non-functional requirement to be analyzed and *Topic* represents some characteristic concerning the application to which *Type* is applied. Fig. 1 illustrates an example of NFR softgoals (graphically represented as clouds), with their respective interconnections. The example will be described throughout the rest of this paper.
2. A catalogue of topic decomposition methods, consisting of:

   - *Decomposition of the software system by user's operational profile.* This decomposition employs an approach similar to that adopted in [6], with the availability of a system being initially analyzed in accordance with the user's operational profile. In the example, this decomposition receives the name *WebSoftwareSystemAvailabilityViaProfile*, and it is applied to two operational profiles: *Simple Profile* and *Special Profile.* The decomposition employs an AND contribution, which means that in order for the softgoal representing the availability of a software system to be satisficed, all availability softgoals related to operational profiles must be considered satisficed;

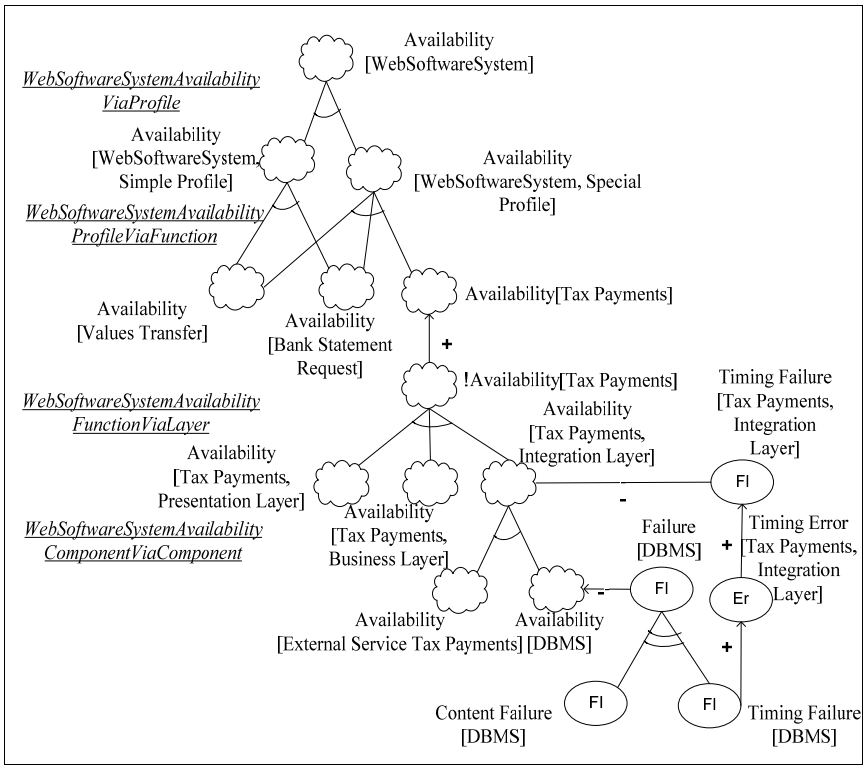

Fig. 1. Graph featuring availability generated by the WSSAA technique.

- *Decomposition of the software system in terms of functions.* The softgoals representing the availability of a certain user's profile may be decomposed, by topic, into softgoals that represent the availability of the functionalities to which the users associated to the profile have access. In figure 1, the decomposition has the title *WebSoftwareSystemAvailabilityProfileViaFunction*;
- *Decomposition of system functionality into components.* According to [8], the structure of the system enables the system itself to be capable of generating the behavior observed by a user, with the current state of the system being determined by the set of its external components. Based on these definitions, a decomposition by topic of function through their components is proposed. In the case of the class of e-business software systems covered in this article, the first decomposition based on components considers the division into layers (presentation, business rules and integration) and each one corresponds to a component to be analyzed in the system. In the example, decomposition by topic *WebSoftwareSystemAvailabilityFunctionViaLayer* occurs;
- *Decomposition of the component into other components.* According to [8], a system corresponds to a set of components bound together in order to interact, wherein each component may be considered another system. The recursion of this definition stops when one component may be considered atomic, in other

words, when there is no possibility or interest to discern a new internal structure. In the WSSAA technique, decomposition by topic is established, considering the system structure. In the example, the decomposition *WebSoftwareSystemAvailabilityComponentViaComponent* exists with a contribution AND, justified by the fact that the execution of the service of a component depends on the execution of *the sub-components involved;*

3. *Prioritized availability perceived by a certain user's profile or system function.* In the context of software systems with distinct operational profiles, business rules may determine the priority of availability pertaining to a certain user profile or functionality. The WSSAA technique adopts the same notation employed for the prioritization of NFR softgoals on the NFR framework: a positive contribution of the prioritized softgoal, with the character '!' accompanying the cloud where it is represented. Figure 1 illustrates an example of prioritization of the availability *softgoal* that has as the functionality *Tax Payments* as a topic.

4. *Entities that represent threats to availability (faults, errors and failures).* As defined in [3], the construction of availability scenarios considers faults as stimuli, classified according to the taxonomy of failures presented in [1]. In order to generate scenarios that adhere to the definition of dependability in [8], a representation of threats to dependability in the WSSAA technique is necessary. As there is no representation of failures in the NFR framework, a new element to represent threats to dependability must be created. As the softgoal concept concerns the goals to be achieved in a system, the softgoal cannot be employed to represent threats to dependability, because the threats do not correspond to the goals to be achieved. Therefore, the creation of a new element is proposed, the threat to dependability. According to the definition in [8], there are three types of threats to dependability: fault, error and failure[1]. Each threat is represented by an ellipse, with initials identifying each type of threat. The initials *Ft*, *Er* and *Fl* represent fault, error and failure, respectively. In the proposed technique, the elements that represent threats to dependability, like NFR softgoals, present a description *Type[Topic]*, in which *Type* represents a threat to dependability, as described in [1] and [8], and *Topic*, corresponds to the element associated to the threat described by the *Type*.

5. *Failures decomposition by type.* According to the classification of failures by domain described in [1], failures may be classified as *content failures***,** where the content of the information delivered at the service interface deviates from the correct information; *timing failures*, where the response time concerning the service is not in accordance with the specification; and *content and timing failures***,** where the failure is associated to time and value factors. According to this classification, the WSSAA technique proposes failure decomposition methods. One example is illustrated in Figure . These decompositions essentially employ *OR* contributions, indicating that the occurrence of any type of failure characterizes a failure concerning a certain topic. There are several possibilities for representing

[1] Failure can be seen as an event occurs when the delivered service deviates from correct service; error is a part of system state which is liable to lead to failure and fault is the adjudged cause or hypothesized cause of an error [8].

negative contributions from these failures and several NFR softgoals; for instance, between timing failure and response time softgoals, or between content failure and accuracy softgoals, among others. Although the representation of these contributions is important for the characterization of failures, they will not be employed here in the generation of availability scenarios.

6. *Contribution between failures and availability softgoals.* According to the definition in [8], attributes of dependability are affected by threats to dependability, i.e., the occurrence of failures negatively contributes to the availability. Thus, the technique proposed adopts the interdependency between the concept of failures and the availability softgoal by means of a *HURT* contribution (if a derived softgoal is satisficed, the parent softgoal may be partially denied) [4]. The employment of a *HURT* contribution is based on two premises. The first consists of the fact that the occurrence of a failure in a topic may have negative effects on the availability softgoal of this topic. The second concerns the intensity of the contribution, because we consider that the occurrence of a failure in a topic indicates that the availability softgoal may possibly be considered unsatisfactory. But, in this case, it will not necessarily be unsatisfactory, since possible operationalizations concerning means of fault tolerance may result in an availability softgoal presenting a failure being considered satisfactory.
7. *Contributions among faults, errors and failures.* [1] and [8] have established a fundamental chain of threats to dependability, indicating a threat mechanism creation and manifestation. In summary, faults activate errors, which propagate into failures, hence causing further faults. This fundamental chain of threats will be represented in the proposed method through contributions of the *HELP* type (if a derived softgoal is safisficed, the parent softgoal may be partially satisficed) [4], i.e., characteristics that aid in the linking of threats. However, this linking may be avoided by adopting fault tolerance means.

The proposed technique consists of two main stages. In the first stage, a graph is generated containing NFR softgoals, threats to dependability and interdependencies between these elements. In the second stage, the application of an evaluation procedure enables the identification of elements resulting in the creation of availability scenarios for architectural evaluation.

The first stage begins with an analysis of system availability and consists of the following steps:

1. The decomposition *WebSoftwareSystemAvailabilityViaProfile* should be applied;
2. In the event that some operational profile requires a greater level of availability from the system, priority marking must be employed.
3. The decomposition *WebSoftwareSystemAvailabilityProfileViaFunction* must be applied based on the functions of each operational profile.
4. In the event that some function requires a higher level of availability compared to the other system functions, priority marking must be employed;
5. Application of the decomposition by topic *WebSoftwareSystemAvailabilityFunctionViaLayer*, considering the presentation, business rules and integration layers;

6. Reiterated application of the decomposition by topic *WebSoftwareSystemAvailabilityComponentViaComponent* for the identification of the components to be analyzed;
7. Analysis of possible failures that may occur in the available services for these components. The failure decomposition methods by type, defined according to the failure domain classification, aid in the identification and characterization of possible component failures.
8. The characterization of each failure detailing the connections between failures and availability softgoals.
9. Representation of the negative contributions that may occur between failures of components and their respective availability softgoals.
10. Identification of threats to dependability that may be the consequence of the occurrence of failures in other components.
11. As these failures are being identified in system components, and as long as they affect the availability of a certain component, negative contributions of the availability softgoal associated with this component must be plotted on a graph.

The result of the first stage is, therefore, the creation of a graph of interdependencies among NFR softgoals and threats to the dependability of the software system.

The second stage of the technique is based on the evaluation procedure described in the NFR framework [4] applied to the obtained interdependency graph and aims to qualitatively evaluate the relationship between failures, that is, stimuli of availability scenarios, and the possible responses of the system associated to the availability softgoals. In the proposed technique, it is considered that component failures will serve as stimuli for the generation of availability scenarios. As component failures may be monitored, the results of this monitoring consist of system responses in the occurrence of those stimuli. Accordingly, there is the possibility of finding, for each possible failure identified, which components, operational profiles and functions may have their level of availability compromised. This finding enables the establishment of desired levels of availability in the occurrence of the identified stimuli, resulting in an availability scenario. On the other hand, the softgoals pertaining to availability associated to operational profiles, functionalities and components are related to these stimuli. In summary, the second stage corresponds to the application of an evaluation procedure, aimed at the creation of a catalogue that gathers all the elements employed in the generation of availability scenarios for the system under analysis.

The original goal of the evaluation procedure described in [4] is to determine the degree to which non-functional requirements are achieved by a set of decisions, represented by operationalizing softgoals. In order to fulfill such a goal, the evaluation procedure assigns labels to the softgoals (✓ for satisficeable, X for deniable, C for conflicting and U for undetermined), and applies a label-propagation algorithm, according to the contributions among softgoals.

In the WSSAA technique, the evaluation procedure must employ labels and a propagation algorithm to relate threats to dependability to availability softgoals. At the beginning of the procedure, the failure to be analyzed as a stimulus is marked with

the label ✓. The label propagation algorithm, applied in accordance with the interdependencies present on the graph, enables the identification of which threats are linked to and which NFR softgoals are affected by the threat. Finally, for the analyzed failure, the availability softgoals marked with the label X and the threats marked with the label ✓are connected to the failure under analysis. This information is then grouped into a catalogue, whose format for the example is partially illustrated in Table 1.

Table 1. Table representing failures with softgoals and related threats.

Stimuli	Availability Softgoals	P	Linked Threats
Timing Failure[DBMS]	Availabilty[Tax Payments, Integration Layer]		Timing Error [Tax Payments, Integration Layer]
	Availability[Tax Payments]	✓	Timing Failure [Tax Payments, Integration Layer]
	Availability[Special Profile]		

The first column (Stimuli) represents possible stimuli. The second column (Availability Softgoals) represents the availability softgoals affected by the stimuli. These softgoals serve as a basis for the definition of system responses for availability scenarios. The third column (P) is directly related to the second column, and marks in this column indicate that the softgoals of the second column are marked as priorities on the graph. This column aids in the prioritization of availability scenarios. Finally, the fourth column represents threats related to the existing threat in the first column.

Therefore, the result of the second stage of the method is a table with all the failures and their connections, constituting a catalogue for the generation availability scenarios. The architect should employ this catalogue in step 4 of the ATAM for generating availability scenarios, which must be prioritized and included in the ATAM utility tree. It is worth emphasizing that the quantification of availability levels present in the response of the scenarios and the prioritization of availability scenarios in comparison with scenarios related to other quality attributes are not part of this technique.

4 Final Considerations

In this paper, the WSSAA technique was proposed, a technique that aims to aid in the generation of availability scenarios, a known complex task, in e-business systems for the architecture evaluation method ATAM. The technique is not intended to cover all the possibilities of generation of availability scenarios, but rather to provide, in a detailed and structured manner, the creation of a catalogue enabling the identification and characterization of possible failures that may affect the availability of the system, and enabling the identification of possible consequences in the presence of these failures.

To fulfill this goal, the technique makes use of the dependability concepts covered in [1] and [8], structured with elements present in the NFR framework [4] and

employing the hierarchy proposed in [6] for availability modeling in e-business systems.

One related work is the QAW (Quality Attribute Workshop) [2], a technique for the generation of quality attribute scenarios. The QAW is a method that engages system stakeholders early in the life cycle of software development to generate, prioritize, and refine quality attributes scenarios before the software architecture is completed. The scenario generation procedure in the QAW is essentially based on the stakeholders' experience with the system. Therefore, the generation of scenarios may be compromised since important scenarios concerning non-functional requirements may not be considered. The WSSAA technique helps to minimize the risk pertaining to the availability quality attribute by executing, in a systematic fashion, structuring and characterization procedures involving elements employed in the generation of availability scenarios for the system under analysis.

In a future work, the WSSAA technique will be applied to a case study, involving software architectures that provide e-business services and trying to measure the efficiency of this technique relatively to other ATAM evaluation techniques. Another future work will develop a tool to aid the generation of availability scenarios, through the automation of the steps described in the WSSAA technique. In future studies, work is intended on two extensions of the technique. One of them is to extend the field of application to software architectures different from the architecture presented in this article. The other is the insertion of operationalizing softgoals to represent architectural tactics that cover means of dependability (especially fault tolerance), aiming at aiding the analysis of architectural tactics based upon the scenarios generated in step 5 of the ATAM.

5 References

1. Avizienis A., Laprie J.C., Randell B., Landwehr C.: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11--33 (2004)
2. Barbacci, M., Ellison, R., Stafford, J.A., Weinstock, C.B., Wood, W.G.: Quality Attribute Workshops (QAWs), 3rd Edition. Technical Report CMU/SEI-2003-TR-016 (2003)
3. Bass, L., Clements P., Kazman R.: Software Architecture in Practice, Second Edition. Addison-Wesley (2003)
4. Chung L., Nixon B.A., Yu E., Mylopoulos J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers (2000)
5. Clements, P., Kazman R., Klein M.: Evaluating Software Architectures: Methods and Case Studies: Addison-Wesley (2002)
6. Kaaniche M., Kanoun K., Rabah M.: A Framework for Modeling Availability of e-Business Systems. In 10th International Conference on Computer Communications and Networks, pp. 40--45 (2001)
7. Kazman R., Klein M., P. Clements.: ATAM: Method for Architecture Evaluation. Technical Report CMU/SEI-2000-TR-004 (2000)
8. Laprie J. C.: Dependability: Basics Concepts and Terminology in English, French, German,Italian and Japanese. In Dependable Computing and Fault-Tolerant Systems – Vol 5. Spring-Verlag (1992)
9. Al-Khateeb W.F., Al-Irhayim S., Al-Khateeb, K.A.: Reliability Objectives in Next-generation Internet.In 9th Asia-Pacific Conference on Communications, pp 192--197. (2003)