Reliability Analysis of Concurrent Systems using LTSA

Genaína Nunes Rodrigues*, David Rosenblum
Department of Computer Science
University College London, UK
{G.Rodrigues,D.Rosenblum}@cs.ucl.ac.uk

Jonas Wolf
Institute for Pervasive Computing
ETH Zürich, Switzerland
wolfj@inf.ethz.ch

1. Introduction

The analysis for software dependability is considered an important task within the software engineering life cycle. However, it is often impossible to carry out this task due to the complexity of available tools, lack of expert personnel and time-to-market pressures. As a result, released software versions may present unverified dependability properties subjecting customers to blind software reliability assessment. In particular, concurrent systems present certain behaviour that require a more complex system analysis not easily grasped at system design and architecture level.

In previous work, we defined a technique to predict software system reliability based on scenario specifications [3]. The technique relies on LTSA, the Labelled Transition Systems Analyser tool [4], which provides scenario-based model synthesis and model checking capabilities to support the analysis. In a follow-up work [2], we devised a UML profile for reliability analysis by extending the UML 2.0 specification to support reliability prediction using our LTSA-based approach. In doing that, we have defined basic mechanisms to structure models consistently and to express formally the semantics of the model in a UML-standard way.

In this demo, we present an extension of LTSA that facilitates our approach to reliability prediction. LTSA allows the use of behavioural models for distributed systems as prototypes to explore system behaviour, and automated checking of model compliance to properties (i.e., model checking). To support reliability prediction, our new extensions allow annotating a scenario specification with probabilities and using LTSA to process the resulting scenarios. In this context, scenarios are partial descriptions of how components interact to provide system functionality. A scenario specification is formed by composing multiple scenarios possibly from different stakeholders. The enhanced tool then compiles these annotated scenarios by applying stochastic Finite State Process (FSP) generation. Finally, our new LTSA plugin performs reliability analysis from the generated probabilistic architectural model.

The advantages of using LTSA for reliability analysis are twofold. Firstly, the tool enables the system stakeholder to model the system closer to their own perspective instead of requiring the users to provide very fine grained state machine models of the system. Secondly, it provides the ability of identifying interactions that arise from the concurrent nature of the system's behaviour, the so-called implied scenarios [5], which can potentially decrease the system's reliability. These two benefits together make LTSA a suitable tool for reliability analysis. However, the purpose of the tool is not to model all the component interactions of a system. The main aim is to provide a bird's eye view of the system focusing on reliability, and to identify components and scenario transitions with higher impact on a system's reliability by means of sensitivity analysis.

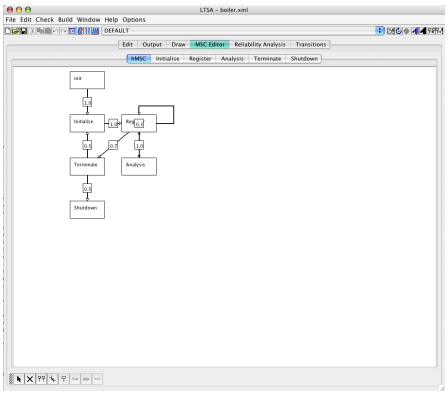
2. Tool Environment and Demo

In Figure 1 we depict some screenshots of the major environments involved in the reliability analysis in LTSA. The process of reliability computation starts at modelling the HMSCs (Figure 1(a)), and the annotation of probability information on those scenario transitions representing choices. The next step is the modelling of the BMSCs (Figure 1(b)), and the annotation of the components with their reliabilities. After the modelling and annotation step is concluded, the construction of the probabilistic architectural model can be performed. Figure 1(c) depicts the synthesized LTS with probability information on the transition between states. The following step is to use the reliability plugin to compute the system reliability from the synthesized architectural model.

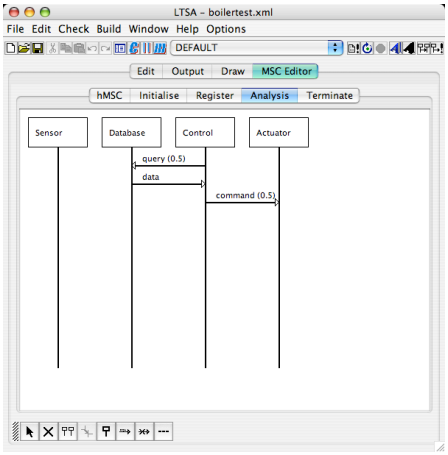
We demonstrate our reliability analysis extension to

*Funded by CAPES, Brazil

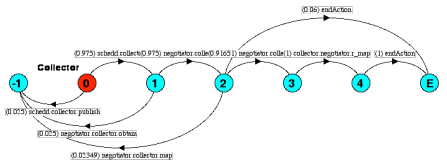
29th International Conference on Software Engineering (ICSE'07 Companion)
0-7695-2892-9/07 $20.00 © 2007 IEEE



(a) An Annotated HMSC Scenario



(b) A Annotated BMSC scenario



(c) A Component Final Probabilistic Composed Model

Figure 1. Reliability Computation Using LTSA

LTSA using two examples: one toy example and one real system used as a case study. Our toy example, is the Boiler system we used in previous papers to illustrate the extended LTSA to support reliability analysis. This includes automated support for sensitivity analysis in order to identify those components and scenario transitions that have the highest impact on system reliability.

Our case study is the Condor system [1], a distributed job scheduler and resource management system, which provides high throughput computing. In order to guarantee high availability of resources, Condor extensively uses the *fail over* mechanism, which is essentially fault tolerance via replication of the various CPU resources to execute jobs. As soon as a resource fails, the job execution is immediately transferred to another available resource until execution finishes. We show how our analysis captures the impact of fail over on the reliability of Condor, and reveal how we abstracted the intricacies of modelling such a complex system to analyse its reliability.

The demonstration of both examples proceeds according to the following steps:

1. Scenarios annotation.
2. Scenarios compilation and synthesis of probabilistic architectural model.
3. Reliability computation.
4. Implied scenarios verification.
5. Synthesis of the new architectural model, now taking into account the implied scenarios.
6. Finally, reliability computation of the new architectural model and comparison of the results from the first and the second runs.

References

[1] C. Chapman, P. Wilson, T. Tannenbaum, M. Farrellee, M. Livny, J. Brodholt, and W. Emmerich. Condor services for the global grid: Interoperability between condor and ogsa. In *Proceedings of the UK E-Science All Hands Meeting, Nottingham*, 2004.

[2] G. Rodrigues, D. Rosenblum, and S. Uchitel. Reliability prediction in model driven development. In *In Proc. of ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, LNCS 3713*, pages 339 – 354. Springer, Oct. 2005.

[3] G. Rodrigues, D. Rosenblum, and S. Uchitel. Using Scenarios to Predict the Reliability of Concurrent Component-Based Software Systems. In *Proc. ETAPS 2005 Conference on Formal Approaches to Software Engineering*, pages 111–126. Springer, LNCS 3442, 2005.

[4] S. Uchitel, R. Chatley, J. Kramer, and J.Magee. LTSA-MSC: Tool Support for Behaviour Model Elaboration Using Implied Scenarios. In *Proc. of* 9^{th} *TACAS, Warsaw*, Apr. 2003.

[5] S. Uchitel, J. Kramer, and J.Magee. Incremental Elaboration of Scenarios-Based Specifications and Behavior Models Using Implied Scenarios. In *ACM Transactions on Software Engineering and Methodologies*, volume 13(1), pages 37–85. ACM Press, Jan. 2004.