Model-based Evaluation for Dependability Assessment of CAUTION++ Instances

Felicita Di Giandomenico[1] Stefano Porcarelli[1], Daniela Viva [1], Andrea Bondavalli[2], Paolo Lollini[2]

[1]Italian National Research Council, ISTI Dept., via Moruzzi 1, I-56134, Italy
{digiandomenico, s.porcarelli, d.viva}@isti.cnr.it
[2]University of Florence, Dip. Sistemi e Informatica, via Lombroso 67/A, I-50134, Italy
{lollini, a.bondavalli}@dsi.unifi.it

Abstract

With reference to the European project CAUTION++, this paper addresses dependability analysis of the CAUTION++ architecture, specifically focusing on the instance chosen for the demonstrator involving GSM/GPRS and WLAN network technologies. The emphasis is on components correctness and reliability issues, which unavoidably need to be addressed to some extent to cope with malfunctions in such complex environment. We apply a modelling technique based on Petri nets in order to model and analyze the behavior of the chosen CAUTION++ instance. The utility of such study consists in a deep understanding of the impact of the correctness of the single architecture's components on the overall dependability of the CAUTION++ system, as well as the impact of fault tolerance measures, introduced to enhance system correctness.

1 Introduction

The challenge of resource management and mobility support, especially in multiple radio environments as pursued by the IST-2001-38229 CAUTION++ [4], unavoidably results in a higher system complexity that deserves special attention. In fact, CAUTION++ aims at developing a novel, low cost, flexible, highly efficient and scalable system able to be utilized by mobile operators to increase the performance of all network segments. To pursue such goal, proper system components are developed to handle generated alarms through a set of RRM (Radio Resource Management) techniques, to be applied where needed. The implication is that issues concerning the dependability [1] of the components/mechanisms composing the resource management architecture need to be addressed to some extent. In fact, behavior correctness, reasonably attained when dealing with simple system components, becomes hard to achieve when complex functionalities are introduced, which have to cope with a variety of external and internal system behaviors. For these reasons, the CAUTION++ project has promoted model-based evaluation, aiming at assessing dependability attributes of the architecture under development. Given the hierarchical structure of the system components (ITMU, RMU and GMU), a hierarchical and modular modeling methodology has been defined to efficiently cope with the system complexity. This methodology, based on an extension of the Petri net formalism and already presented in [3], will be briefly recalled later on. To better tailor the proposed methodology to the CAUTION++ project and demonstrate its efficacy in the related context, in this paper we consider a specific architecture's instance involving GSM/GPRS and WLAN network technologies deployed by two distinct operators, which is actually one of the demonstrators chosen by the consortium to show the project's results. The utility of our study mainly consists in a deep understanding of the impact of the correctness of the single architecture's components on the overall dependability of the CAUTION++ system, as well as the impact of fault tolerance measures, introduced to enhance system correctness.

The rest of this paper is organized as follows. Section 2 presents the CAUTION++ instance considered in the analysis. Section 3 briefly introduces the adopted modeling methodology. In Section 4 the models set-up for the selected CAUTION++ instance are discussed, while the results of the numerical evaluation are provided in Section 5. Finally, conclusions are in Section 6.

2 The analyzed CAUTION++ instance

In order to provide a solid proof of the CAUTION++ concepts, a few demonstrations have been planned as part of

the project's technical activity, consisting of trials and simulations. Specifically, trials are devised to prove some scenarios of system utilization, whereas simulations are devoted to examine some other scenarios which cannot be made available within the scope of the CAUTION++ project. The dependability analysis we carry on in this paper focuses on the trial performed by the two partners NTUA and COSMOTE, having the objective of evaluating monitoring and radio resource management for WLAN and GPRS networks. In this trial, the CAUTION++ system is exploited to manage both vertical handover between WLAN and GPRS, as well as roaming between operators.

From the point of view of system composition, Figure 1 depicts the components included in such trial. Three operators are involved, Op1, Op2 and Op3, with Op1 and Op3 managing a WLAN network only, and Op2 managing both a GPRS and a WLAN network. From the point of view of CAUTION++ components employed in this instance, each network segment has its own ITMU (Interface Traffic Monitoring Unit) and RMU (Resource Management Unit) which allow to monitor and manage the attached network, respectively. Within each operator network, a GMU (Global Management Unit) is necessary to perform a global optimization. In fact, different GMUs cooperate to optimize among different operators. Therefore, this CAUTION++ instance includes 4 ITMU, 4 RMU and 3 GMU, connected as shown in the Figure.

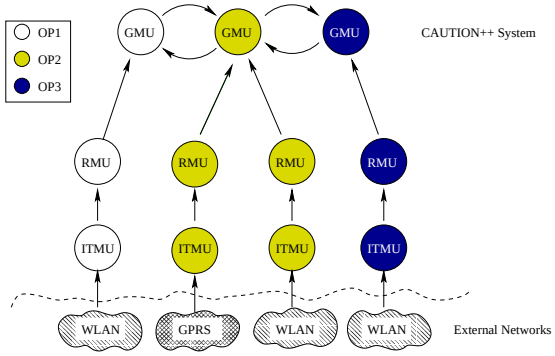

Figure 1. Trial Configuration

3 Outline of the modeling methodology

The CAUTION++ architecture should allow putting in place correctly the identified Resource Management Techniques, hopefully despite the occurrence of faults. Thus, the design of the CAUTION++ architecture necessitates validation and verification activities to be performed as soon as possible from the very early phases of the design process, in order to justifiably trust the identified solutions and to make appropriate choices among several possible alternatives. Model-based validation is promoted inside the CAUTION++ framework to contribute to this purpose.

To cope with system complexity, we resorted to a modular and hierarchical modeling methodology, following a top-down approach. First, an "abstract model" is defined, where the high level behavior of the overall system is captured. From this abstract model, we perform a decomposition in more elementary but more "detailed sub models". Then the model solution is carried out in a bottom-up fashion, from the detailed models solution to the overall system solution through the abstract model. This methodology is deeply described in [3].

In accordance with this methodology, an "abstract" and a "detailed" models are associated with each ITMU, RMU, and GMU. The detailed models take into account the detailed internal behavior of the component (e.g., of its subcomponents and of the fault tolerance mechanisms included in, as discussed in the next subsection). The resolution of such detailed model allows to determine the four probabilities of correct/incorrect emission and correct/incorrect omission. Through the composition of the single "abstract" models, the model of the overall system is obtained, simple enough to be solved through an analytical approach. Note that in our context the flow of computation moves from the ITMU to the RMU and finally to the GMU. Therefore, the model solution of a certain component in this chain determines the probabilities of correct/incorrect input for the next component.

3.1 Components behavior and modeling assumptions

In order to set up the detailed models, a characterization of the system components from the dependability point of view is necessary, briefly outlined in the following.

- Each CAUTION++ element (ITMU, RMU, GMU) can be either correctly working or wrongly working.
- Each CAUTION++ element (ITMU, RMU, GMU) is composed by three main elements: the Application Software (AS), the Operating System (OS), and the Hardware (HW). While the OS and the HW are off-the-shelf components, having dependability figures typically provided by manufactures, the Application Software is the software specifically implemented for CAUTION++. In turn, the AS, OS, and HW can be either correctly working or wrongly working.
- At the end of its computation, each CAUTION++ component can emit an output or not. More precisely, the

possible output can be either correct/incorrect emission or correct/incorrect omission.

- Fault tolerance mechanisms are in place in each system component, in order to improve the dependability of the components themselves and limit the error propagation between interacting elements. They are interface checks (to detect errors at input/output level), diagnosis and repair mechanisms. Their ability to work properly depends on their respective coverage.

In addition, a set of assumptions has been identified with the aim of enhancing simplicity and clarity (essential to keep the whole modeling activity under control), still capturing the relevant phenomena which impact the measures under analysis (essential to the practical usefulness of the evaluation effort). The complete list is in [3]; here we omit those strictly related with details of the models not shown in this paper.

- The input to the detailed model may be either correct with probability α or incorrect with probability 1-α.
- Each CAUTION++ element (ITMU, RMU, GMU) can generate by itself spurious outputs (that is, outputs not triggered by an external input; it is a manifestation of a fault in the component). Spurious outputs are independent from outputs generated by real inputs and follow an exponential distribution with rate $MTBFA$.
- The coverage of the Input interface checks is given by the probability *inputCoverage*. When Output interface checks are considered, the detection of an erroneous output leads to an output omission (correct or incorrect, depending from the inputs originating it and/or the correctness of the component's status) with probability *outputCoverage*.
- An undetected error sooner or later propagates and reveals itself.
- A repair of the OS may allow detecting an undetected error at the AS level (e.g. as consequence of rebooting). In this case, a possibly undetected erroneous state of the AS becomes detected.
- A repair of the HW may allow detecting an undetected error either at the AS or OS level (because of necessary system reboot - no hot-pluggable redundancy is envisioned). In this case, a possibly undetected erroneous state (either of the AS or OS) becomes detected.

4 Sketch of the models derived for the selected CAUTION++ instance

In this Section, the models derived for the analysis of the selected CAUTION++ instance (depicted in Figure 1) are briefly outlined. First, the measures of interest are described, since they influence the definition of the system models.

4.1 Measures of Interest

The main dependability requirement of CAUTION++ is that it should avoid taking wrong decisions, thus acting worse than doing nothing. Particularly, an omission failure (that is the system does not provide any output when, if correct, it would have emitted one) can be tolerated, since it leads to no benefit from CAUTION++. Emission failure instead (that is, an incorrect output is emitted) can lead the system to act worse than doing nothing, and therefore actions would be required to prevent such failure mode. We have identified the following indicators as significant measures to evaluate the dependability of the CAUTION++ architecture. They are:

- The probability of incorrect emission at level of the GMU employed by a certain operator;
- Mean Time to Failure of the GMU employed by a certain operator;
- Reliability of the whole system(with contributions from all the present GMUs).

They show suitable measures to evaluate the ability of CAUTION++ in fulfilling the general dependability requirement of not undertaking wrong reconfiguration actions.

4.2 Detailed and abstract models

In accordance with the proposed methodology, the starting point is the definition of an "abstract" model for each involved component. The generic "abstract" model is represented in Figure 2, using the SAN formalism [5].

It is valid for ITMU, RMU and GMU. The input gate *gInput_X* allows handling the input of the component (both the correct and incorrect input), transition *lambda_X* fires with a rate given by the rate of messages in input to component X. Possibly, an output is produced, which can be either correctly emitted (a token is moved in place *Correct_X*), or incorrectly emitted (a token is moved in place *Corrupted_X*), or correctly omitted (a token is moved in place *NoOutCorr_X*) or incorrectly omitted (a token is moved in place *NoOutIncorr_X*).

To obtain the parameters of each abstract model, the corresponding detailed models have to be set-up and solved. Therefore, a detailed model is built for each involved component. Since ITMU, RMU and GMU employ the same subcomponents (HW, OS and AS, plus fault tolerance mechanisms, as already discussed), the detailed model is

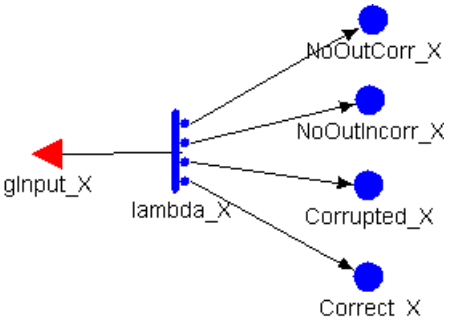

Figure 2. Generic Abstract Submodels

almost the same for all of them. The only difference is in the values of their parameters (as explained later in the section on numerical evaluation). A generic detailed model is obtained by composing the generic detailed models for the component's subcomponents (i.e., HW, OS and AS) together with the dynamics of the error and fault detection mechanisms employed. The presentation of this model is omitted for brevity (refer [3] for a complete exposition); here only the generic detailed model for the subcomponent Y (where Y maybe AS, OS or HW) is sketched in Figure 3.

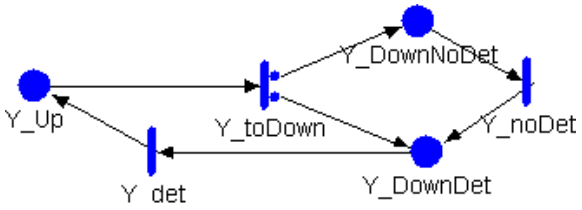

Figure 3. Detailed Model for AS, OS, and HW

A token in place *Y_Up* indicates that *Y* is working correctly. The firing of transition *Y_toDown* models its failure: this failure can be detected (a token moves in the place *Y_DownDet*) or not (a token moves in the place *Y_DownNoDet*) with probabilities *AS_Coverage* and 1-*AS_Coverage*, respectively (*AS_Coverage* represents the coverage of the error detection mechanisms implemented in the Application Software). An undetected failure can be revealed after a while, e.g. through repair mechanisms applied at OS or HW level; the firing of transition *Y_det* indicates such failure detection. A detected failure is then recovered by means of the transition *Y_repair*.

The overall model for the CAUTION++ instance under analysis has been determined under the following assumptions:

- Messages coming from different ITMUs and RMUs are indistinguishable.
- The RMUs and the GMUs process the incoming input requests (from the ITMUs and RMUs respectively) individually and sequentially.

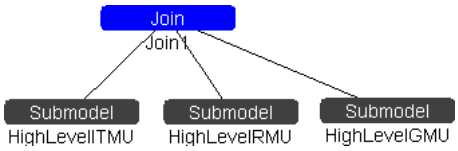

Figure 4. Composed Model at GMU decision level

Figure 4 shows the overall model for analyzing the CAUTION++ behavior at a single GMU decision level (e.g., to evaluate the probability of correctness of a reconfiguration decision issued by a GMU). Thanks to the above assumptions, the evaluation of the whole CAUTION++ instance is easily obtained by mathematically combining the evaluations at single GMU level, in accordance with the specific measure under analysis.

5 Evaluation results

The preceding models have been numerically solved using the analytical solver provided by the Möbius tool (see [2]). Since all the timed transitions are exponentially distributed and the state space dimension of the models was not huge, it was possible to pursue an analytical solution achieving more accurate results than through simulation. Given the nature of the measures of interest, we resorted to a steady-state analysis for all models.

5.1 Settings for the Numerical Evaluation

The developed models have a number of internal parameters, to which values have to be assigned. For many of them, reference values from manufactures or previous studies in the literature are available. For others, mainly those concerning the components to be developed in the CAUTION++ framework, this is not true and the choice of appropriate values is more critical. Therefore, for such critical parameters, a range of values is experimented in the analysis, to determine the impact of such variations on the analyzed dependability figures (sensitivity analysis). Table 1 lists the varying parameters, and the range of values assigned to them in the analysis. The extension *_X* makes the parameter's name generic, and need to be properly substituted by ITMU, RMU, GMU to indicate the parameters of the corresponding component. Since the models have been just sketched in this paper, details of the involved parameters would be very difficult to understand. For them, the same values used in [3] have been applied.

Parameter	Range
α_{ITMU}	0.90 - 0.999
α_{RMU}	from ITMU
α_{GMU}	from RMU
MTBA_ITMU	2- 48(hours)
MTBA_RMU	from ITMU
MTBA_GMU	from RMU
InputCoverage_X	0.00 - 1.00
OutputCoverage_X	0.00 - 1.00
AS_Coverage_X	0.70 - 0.999
MTBFA_X	198 - 2000(hours)

Table 1. Varying Model Parameters and their values

In the table:

α_{ITMU}, α_{RMU} and α_{GMU} are the probabilities that the input provided to ITMU, RMU and GMU, respectively, is correct;

MTBA_ITMU, MTBA_RMU and MTBA_GMU are the mean time between two inputs to ITMU, RMU and GMU, respectively (in the case of ITMU, it is the mean time between two external inputs for which ITMU generates an alarm to RMU);

MTBFA_X is the mean time between two spurious outputs emitted by a generic component X ;

InputCoverage_X is the coverage of the error detection checks at input interface;

OutputCoverage_X is the coverage of the error detection checks at output interface;

AS_Coverage_X is the coverage of the application software checks.

5.2 Numerical Evaluation

In this section, we present and discuss the results obtained.

To keep the notation as much light as possible, in the figures I/OCov is the coverage of the input and output interface (which is the same for ITMU, RMU and GMU), AS-Cov is the coverage of the application software (again, it is the same for ITMU, RMU and GMU).

Figure 5 shows the probability of incorrect emission of the GMU managed by Operator1 (it is actually the same for Operator3 also), at varying values of the coverage of the I/O Interface Checks and the coverage of the Application Software. The probability of incorrect emission decreases as the probability of coverage of the I/O Interface Checks increases; instead, it is not influenced by As Coverage. Looking at the two overlapping curves in the figure, it can be observed that the impact of the correctness of the input to ITMU is not relevant. Therefore concerning the emission failure probability, significant benefits are achieved using the Interface Checks, since more incorrect messages are detected and no output is produced in these cases.

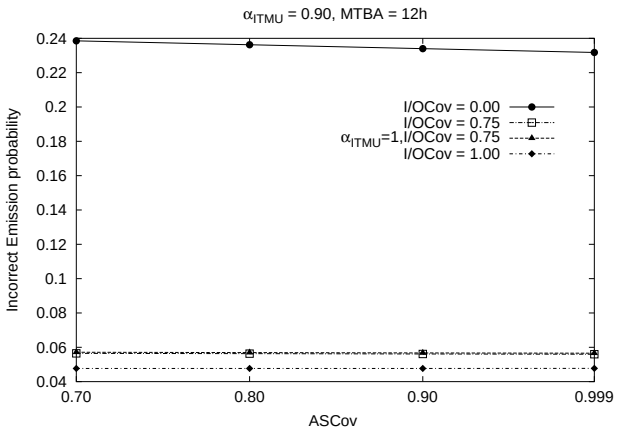

Figure 5. Incorrect Emission Probability related to Operators 1 and 3

Figure 6 and figure 7 are plotted at varying values of the Mean Time Between Alarms and the Mean Time between spurious outputs, and setting to 0.98 the probability that the input to ITMU is correct. Not surprisingly, all the curves follow an increasing trend. Note that the time to an incorrect emission is sigificantly different for Operator 1 (or Operator 3) and Operator 2.

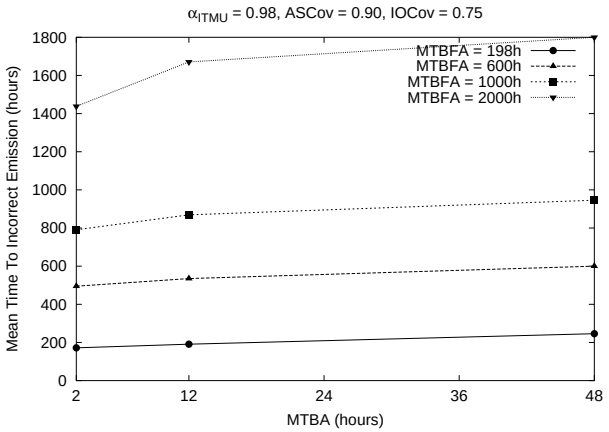

Figure 6. Mean Time To Incorrect Emission for Operator 1 or Operator 3

The last figure shows the reliability of the Trial system at varying the observation time. It has been obtained by fixing the Mean Time Between Alarms to 12 hours and the probability of correct input to ITMU to 0.98. The varying parameter is the MTBA. The Reliability of the system quickly

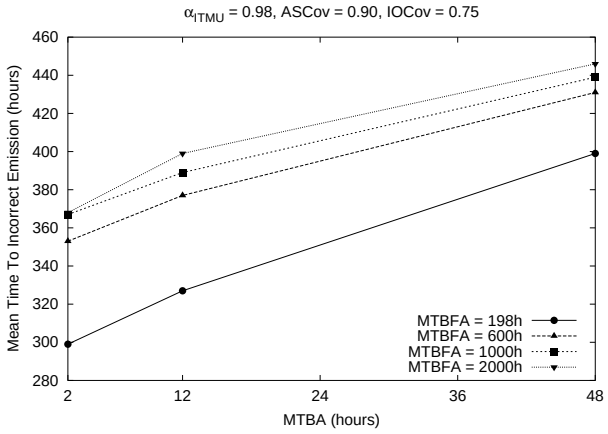

Figure 7. Mean Time To Incorrect Emission for Op2

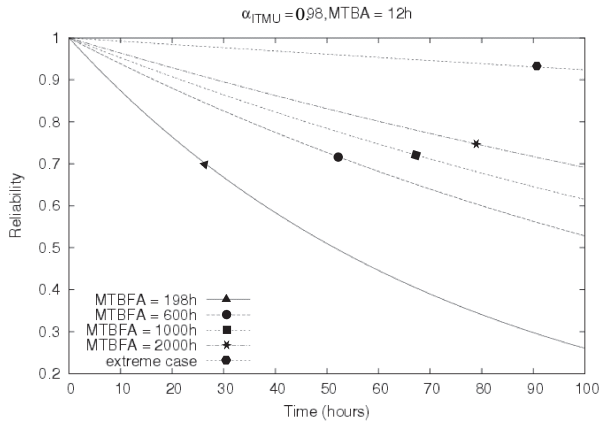

Figure 8. Reliability of the Trial system

decreases at lower values of MTBA. In the figure, also an "extreme case" curve is plotted, obtained considering totally correct the external input to the ITMU, and assuming a very high coverage (0.99) for all the employed error detection mechanisms. The idea was to understand how would be the reliability of the CAUTION++ instance, in case a highly robust implementation of the CAUTION++ components is performed and in absence of faults external to the system. It can be appreciated that in such a case the reliability curve has a very good trend.

6 Conclusions

This paper has presented an evaluation study of dependability indicators in a particular CAUTION++ instance, which corresponds to one of the trials chosen to demonstrate the projects results. A modular and hierarchical methodology has been adopted, both at level of models representation and at level of models solution. This methodology is especially effective to cope with system complexity and state space explosion problem. In accordance with basic dependability requirements stated in CAUTION++, the evaluated dependability indicators have been the probability of an incorrect output emission, the Mean Time to Failure of a GMU component and the reliability of the whole instance. We resorted to an analytical solution, using the automatic Möbius tool. The obtained results allow to understand the impact of several factors contributing to the dependability of the single CAUTION++ components on the overall system instance. This study can therefore be useful to guide implementation choices addressing dependability, by providing comparative quantitative assessment of possible alternatives.

7 Acknowledgments

This work has been partially supported by the European Community through the IST-2001-38229 CAUTION++ project and by the Italian Ministry for University, Science and Technology Research (MURST), project "Strumenti, Ambienti e Applicazioni Innovative per la Societa' dell'Informazione, SOTTOPROGETTO 4".

References

[1] A. Avizienis, J.-C. Laprie, and B. Randell. Fundamental concepts of dependability. Technical Report CS-TR-739, Newcastle University, 2001.

[2] D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders. Möbius: An Extensible Tool for Performance and Dependability Modeling. In *11th International Conference, TOOLS 2000*, volume Lecture Notes in Computer Science, pages 332–336, Schaumburg, IL, 2000. B. R. Haverkort, H. C. Bohnenkamp, and C. U. Smith (Eds.).

[3] S. Porcarelli, F. D. Giandomenico, A. Bondavalli, and P. Lollini. Model-based evaluation of a radio resource management system for wireless networks. In *Computing Frontiers (to appear)*, Ischia, Italy., April 2004.

[4] C. I. Project. Capacity Utilization in Cellular Networks of Present and Future Generation++. http://www.telecom.ece.ntua.gr/CautionPlus/.

[5] W. H. Sanders and J. F. Meyer. A Unified Approach for Specifying Measures of Performance, Dependability and Performability. In *Dependable Computing for Critical Applications*, volume 4 of *Dependable Computing and Fault-Tolerant Systems*, pages 215–237. Springer Verlag, 1991.