MODELING AND SIMULATION FOR THE PERFORMANCE EVALUATION OF THE ON-BOARD COMMUNICATION SYSTEM OF A METRO TRAIN

Alberto Falcone, Alfredo Garro, Andrea Tundis

Department of Informatics, Modeling, Electronics, and Systems Engineering (DIMES), University of Calabria, via P. Bucci 41C, 87036, Rende (CS), Italy

{alberto.falcone, alfredo.garro, andrea.tundis}@dimes.unical.it

ABSTRACT

The paper presents the evaluation of the dependability performances of a real On-Board Communication System of a Metro train centered on the application of RAMSAS, a recently but promising model-based method for the reliability analysis of systems through Simulation. In particular, after the description of the On-Board Communication System under consideration, of its dependability requirements, and related performance indicators to be evaluated, a SysML-based model of the structure and behavior of the system is presented. Beside the nominal system behavior, specific dysfunctional tasks, able to alter the intended behavior of the system, are introduced in order not only to evaluate through Simulation system dependability performances but also to compare different design choices and parameters settings against the requirements.

Keywords: Model-Based Systems Engineering, Safety, Availability, Reliability, Performance Evaluation, Urban Rail Transport

1. INTRODUCTION

Non-functional requirements analysis and related system performance evaluation are challenging tasks that involve several disciplines ranging from Modeling and Simulation to Systems Engineering. These tasks rely on the modeling of system properties that deals with formally expressing constraints and both functional and non-functional requirements so to enable their verification through real or simulated experiments and/or analytical techniques.

Among non-functional requirements, the *dependability* ones (such as *reliability*, *availability*, *maintainability* and *safety)*, which represent important properties to be satisfied for a wide range of systems (Guillerm, Demmou, and Sadou 2010; Laprie 1992; Stapelberg 2008), become really crucial in *mission-critical* industrial domains, such as nuclear plants, avionics, automotive and satellite (Lahtinen, Johansson, Ranta, Harju, and Nevalainen 2010; Rierson 2013; Navinkumar and Archana 2011; Garro, Groß, Riestenpatt Gen. Richter, and Tundis 2013).

As a consequence, international organizations, research centers, and companies are strongly involved in investigation and standardization activities focused on dependability aspects; for instance (i) IEC-61508, provided by IEC (International Electrotechnical Commission), deals with aspects of Electrical, Electronical and Programmable Electronical Systems (IEC-61508 2010) as well as ISO-26262 which represents the reference standard in automotive domain (ISO-26262 2011); (ii) RTCA - DO 254, by the RTCA Special Committee, provides guidance for design assurance of airborne electronic hardware (RTCA/DO 254 2000); (iii) ECSS-Q80-03 is a standard defined by ESA (European Space Agency) concerning methods and techniques to support the assessment of software dependability and safety (ECSS-Q80-03 2006); (iv) NASA/SP-2007-6105 provides top-level guidelines and best practices as well as crosscutting management processes in systems engineering (NASA 2007). These efforts testify the need of models and methods for representing system requirements and constraints able to support their validation, traceability and verification (Krause, Hintze, Magnus, and Diedrich 2012; Peraldi-Frati and Albinet 2010; Tundis, Rogovchenko-Buffoni, Fritzson, and Garro 2013; Yu, Xu, and Du 2009).

Recently, great attention is devoted towards the railway domain and, particularly, on its safety and reliability. Indeed, human errors, as well as deliberate sabotage, pose a considerable danger to passengers travelling on the modern railways and have disastrous consequences. To protect civilians against both intentional and unintentional threats, rail transportation has become increasingly automated and performance studies are fundamental to increase the lifetime of railway systems (Flammini 2012). One of the main goals of this analysis is to verify whether system working conditions are reliable and safe to reduce dangerous situations or even losses of human lives. This task not only takes into account the analysis of the whole traction chain, but also requires ensuring that the railway infrastructure is properly working. As a consequence, several tests for detecting any dysfunctional behavior need to be carried out (Scott, Dadashi, Wilson, and Mills 2013; Reliability and Safety in Railway 2012).

Unfortunately, even though the modeling and simulation of functional requirements are well supported by several tools and techniques, a lack of

methods, models and practices specifically conceived to deal with non-functional requirements and, in particular, with the dependable ones, are still missed; as a consequence, the evaluation of system performances is often delayed to the late stages of the development process with the high risk of having to revise already implemented design choices, and, consequently, to miss project deadlines and budget.

To contribute to fill this lack, the paper exemplifies a comprehensive approach for supporting the evaluation of dependability performances centered on Simulation by taking as the reference system the On-Board Communication System (OBCS) supplied by SELEX (SELEX ES) installed on the Line 5 of the Milan Metro train. Specifically, the experimentation is performed through RAMSAS (Garro and Tundis 2014), a recently proposed model-based method for the reliability analysis of systems through Simulation.

In particular, Section 2 briefly introduces the RAMSAS method, then the description of the On Board Communication System under consideration is provided in Section 3; in Section 4 the dependability requirements and related performance indicators to be evaluated are described. Then the structure and behavior (both nominal and dysfunctional) of the OBCS is modeled in Section 5; whereas, in Section 6, the evaluation of its dependability performances by adopting simulation techniques is presented. Finally, conclusions are drawn and future work delineated.

2. RAMSAS: A MODEL-BASED METHOD FOR DEPENDABILITY ANALYSIS THROUGH SIMULATION

The evaluation of performances of the On-Board Communication System under consideration is performed through RAMSAS (Garro and Tundis 2014), a model-based and simulation-driven method which consists of four main phases: *Reliability Requirements Analysis*, *System Modeling, System Simulation,* and *Results Assessment*. Specifically, in the Reliability Requirements Analysis phase the objectives of the reliability analysis are specified and the reliability functions and indicators to evaluate during the simulation are defined. In the System Modeling phase, the structure and behavior of the system are modeled in SysML (System Modeling Language), the OMG Systems Modeling Language, by using zooming in-out mechanisms (Molesini, Omicini, Ricci, and Denti 2005); moreover, beside the intended system behaviors, specific dysfunctional behaviors and related tasks, which model the onset, propagation and management of faults and failures, are introduced. In the System Simulation phase, the previously obtained models of the system are represented in terms of the constructs offered by the target simulation platform, then simulations are executed so to evaluate the reliability performance of the system also on the basis of different operating conditions, failure modes and design choices. Finally, simulation results are analyzed with respect to the objectives of the reliability analysis; if necessary, new partial or complete process iterations are executed. With reference to a typical V-Model process, RAMSAS can be used: (i) in the testing phases to support the evaluation of unit and system reliability performances; (ii) in the design phases to support the validation and evaluation through simulation of configuration scenarios and setting of system parameters so to evaluate, compare and suggest different design choices and improve the descriptive and predictive capabilities of the reliability system model. RAMSAS has been already experimented in the satellite domain for the reliability analysis of an Attitude Determination and Control System (ADCS) (Garro, Groß, Riestenpatt Gen. Richter, and Tundis 2013), in the avionics domain for the reliability analysis both of a Landing Gear System (Garro, Tundis, and Chirillo 2011) and of a Flight Management System (Garro and Tundis 2012b); and in the automotive domain for the reliability analysis of an Electronic Stability Control (ESC) system (Garro and Tundis 2012a). It combines in a unified framework OMG modeling languages (System Modeling Language) and the popular Mathworks simulation and analysis environments (MATLAB-Simulink).

3. THE ON-BOARD COMMUNICATION SYSTEM

The considered On-Board Communication System (OBCS), supplied by SELEX (SELEX ES) and installed on the Line 5 of the Milan Metro train, is composed by a set of devices required to perform safety tasks and functions as well as the dissemination of information to passengers such as: bidirectional audio communication between the Central Station Operator and Passengers when a situation of emergency occurs, communication between two Central Station Operators, data exchanging for diagnosis among equipment, on board video monitoring, and sending of live/recorded messages to passengers. The general architecture of the OBCS is centered on a Control Unit (CU) subsystem able to manage all system devices and to select the necessary equipment to perform the required functionalities and tasks, as well as to handle, if necessary, voice and data communication by combining Tetra Radio (Terrestrial Trunked Radio) and Wi-Fi components. Figure 1 shows the connections between the CU and the other main communication devices.

Specifically, the components of the considered OBCS are: (i) two Control Unit, (ii) a keyboard for each CU employed in emergency situations by the Central Station Operator on the train, (iii) a Tetra Radio component for each CU for data and audio communication among the Operation Center and the Central Station Operator/passengers on board, and (iv) other components for supporting communication (e.g. Emergency Call Point/ECP, Wi-Fi, Ethernet Switch, Emergency Buttons, Microphones, Speakers, and Amplifiers for Environmental Audio Diffusion).

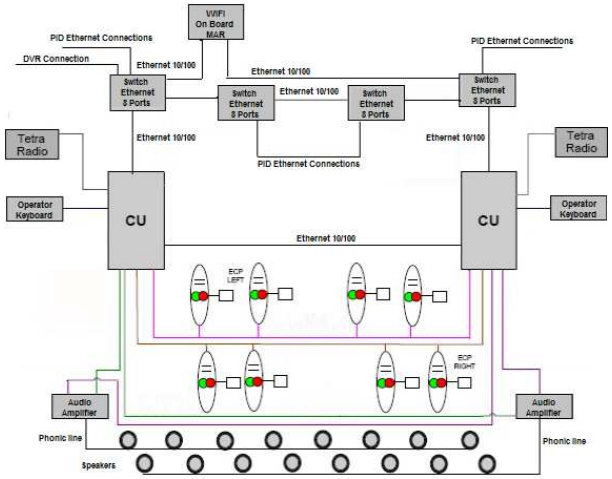

Figure 1: The architecture of the reference On-Board Communication System

4. DEPENDABILITY REQUIREMENTS ANALYSIS

The On-Board Communication System of a Metro train has to fulfill functional and non-functional requirements such as the dependable ones in order to provide a safe mode of transport so as to avoid the occurrence of hazards and to prevent accidents. In general, a system or a component is reliable if it has the ability to perform its required functions under stated conditions for a specified period of time. A more formal definition is based on the concept of MTBF (Mean Time Between Failure) which is defined as the average time that elapses between two successive failures. In this case, the analysis of performance of the system is conducted on the basis of failure events and their effect on the operation of the system. In particular, the following specific fault situations and related performances have been considered:

- Total Block (TB): a block of the system operation for a time more than 3 minutes;
- Partial Block (PB): a block of the overall system operation from 3 minutes up to 10 minutes;
- Delay: a delay of the overall system operation for a time less than 3 minutes.

As it is shown in Table 1, in order to study the behavior of the system subject to the above mentioned type of failures, the following indices are related values have been considered.

Concerning the failures of type TB, the key requirement to be considered is represented by the *Unavailability* calculated as 1-*Availability* which is defined as the ability of an item (e.g. system, subsystem or a component) to perform a required function at a given instant of time or at any required instant of time within a given time interval. In particular, *Availability* is determined according to the equation (1), where the *MPS* represents the minutes of performed service and the *MSS* represents the minutes of scheduled service.

$$Availability = \frac{MSP}{MSS} \quad (1)$$

To study the behavior of the system subject to failures of type PB and Delay, the key requirements to be considered are instead represented by (i) the *Failure Rate* that represents the frequency with which an item (e.g. system, subsystem or a component) fails, expressed, for example, in failures per hour, (ii) the *DownTime_evMAX* that represents the actual duration of the outage resulting from a failure.

Table 1: On-Board Radio System Dependability Requirements

Adverse Event	*Unavailability*	*Failure Rate* (λ) *[hour^{-1}]*	*DownTime_evMAX [hour]*
Total Block	1-0,9999 = 0,0001	-	-
Partial Block	-	2,31E-06	0,16 (~10 min.)
Delay	-	4,33E-06	0,05 (~3 min.)

5. SYSTEM MODELING

This Section describes both the structure and behavior of the OBCS under consideration by exploiting OMG SysML (Systems Modeling Language) as well as zooming in-out mechanisms; moreover, beside the intended system behaviors, specific dysfunctional behaviors and related tasks, which model the onset, propagation and management of faults are introduced in order to fully enable the analysis of the dependability performances through Simulation.

5.1. System Structure Modeling

In this activity a complete (or partial) representation of the structure of the system (or of its parts), that is under analysis, has to be provided. This representation allows for a layered view of the system that is useful to figure out the components involved at some specific layer and the relationships among them. The system structure is modeled by using SysML *Blocks* following a *top-down* approach so as to obtain a hierarchical decomposition of the system (e.g. system, subsystems, equipment, and components). Specifically, each system entity is represented by a SysML *Block* and modeled by both *Block Definition Diagrams* (BDDs) and *Internal Block Diagrams* (IBDs).

As an example, Figure 2 and Figure 3 show respectively the Block Definition Diagram (BDD) and Internal Block Diagram (IBD) relating to the On-Board Communication System, whose components have been already introduced in Section 3 (see Figure 1).

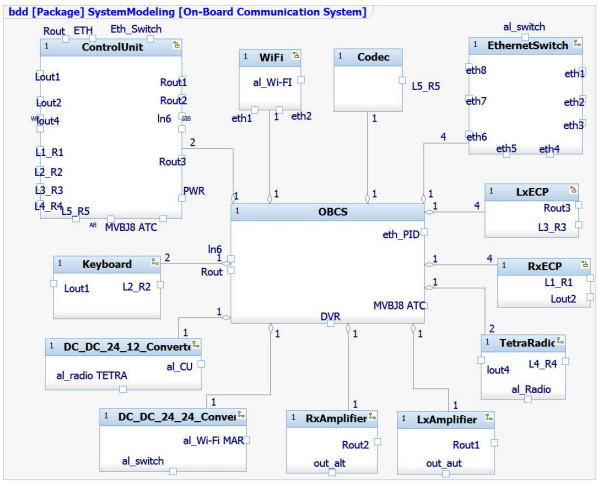

Figure 2: Block Definition Diagram of the On-Board Communication System

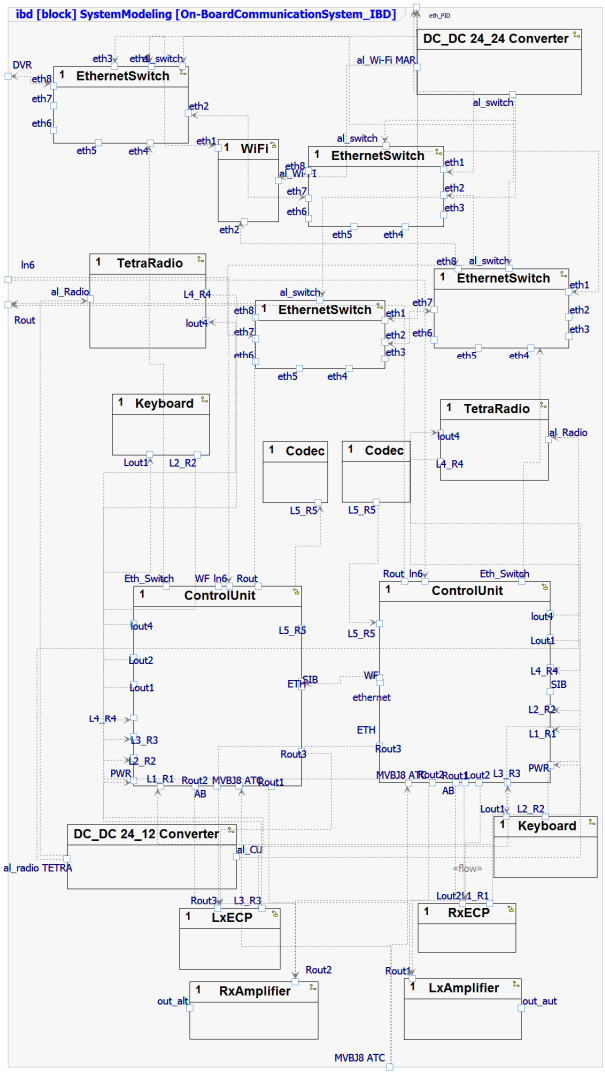

Figure 3: Internal Block Diagram of the On-Board Communication System

By applying *zooming-in* mechanisms (e.g. by breaking down the system) further components can be identified so as to reach a deeper level of decomposition. In the following, the diagrams related to the *ControlUnit* (CU) subsystem are reported; in particular, in Figure 4 and in Figure 5 its BDD and IBD are represented respectively.

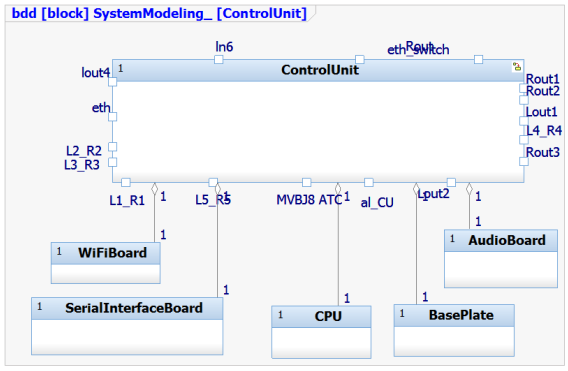

Figure 4: Block Definition Diagram of a Control Unit

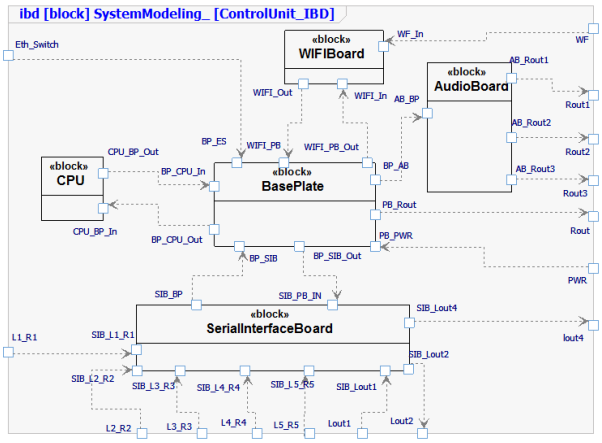

Figure 5: Internal Block Diagram of a Control Unit

5.2. Intended Behavior Modeling

The modeling of the *Intended Behavior* takes into account the hierarchical structure of the system and specifies the intended behavior by following a *bottom-up* approach. Specifically, the behavior of the system entities at the lowest level in the hierarchy, called *leaf level* (e.g. component level), are first specified; then, the behavior of the entities at higher levels of abstraction, called *non-leaf levels* (e.g. equipment and subsystem level), are modeled by specifying how the enclosed entities participate and determine the behavior of each considered enclosing entity.

Different kind of SysML diagrams can be exploited to model the behavior of a given entity: *Activity*, *Sequence*, *Parametric*, and *Statechart Diagrams* according to the characteristics of the behavior and the abstraction level to represent.

In particular, Figure 6 shows the Intended Behavior of a *ControlUnit*, by using a *Statechart,* which is able to handle simultaneously several input signals. Indeed, it can receive and then manage different kinds of signals such as *Passenger Call*, *ForwardMessage, CentralStationOperator-Passenger Call, OnBoardOperator Call, CentralStationOperator-OnBoardOperator Call.* Specifically, when the *CU* is

Active, all its parallel sub-states are in *InWaiting* (see Figure 6).

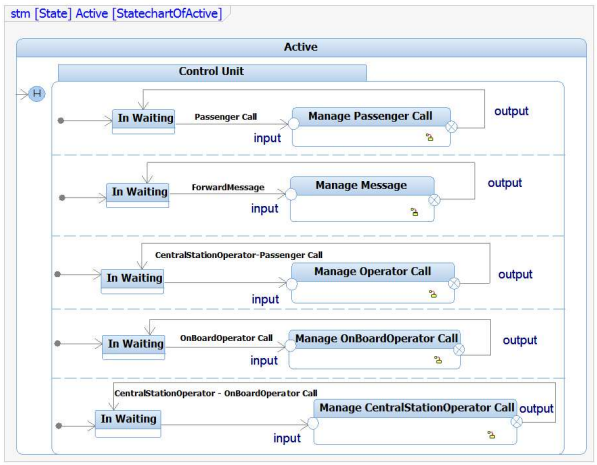

Figure 6: Intended Behavior of a Control Unit

As an example, when a *PassengerCall* is incoming then the behavior of the CU changes its state from *InWaiting* to the *MenagePassengerCall* state that takes in charge such input signal and handles it opportunely by producing a proper output signal.

Figure 7 shows the internal statechart of the *ManagePassengerCall* which aims to initiate, maintain and terminate the call from the passenger to the control center operator. This state is composed by three sub-states: *CallManagement*, which takes care of assigning priority to the call (information/emergency) and to retrieve the identifier of the train where the passenger is located; *ControlCenterConnection* that performs the connection to the control center; finally, the *InCalling* state where resources are allocated to allow communication between the involved parties.

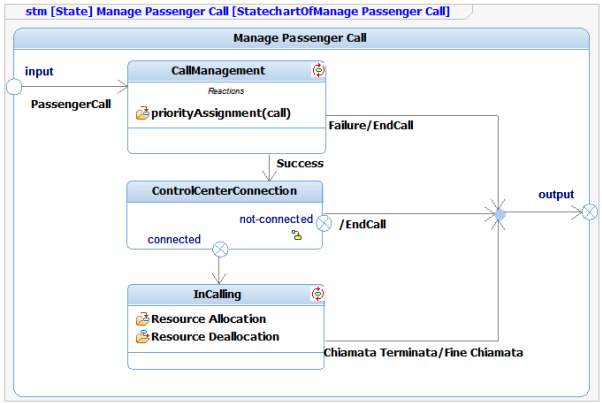

Figure 7: Internal state of the *ManagePassengerCall* of a Control Unit subsystem behavior

A specific path scenario of the above described behavior is shown in Figure 8 by exploiting a SysML Sequence diagram.

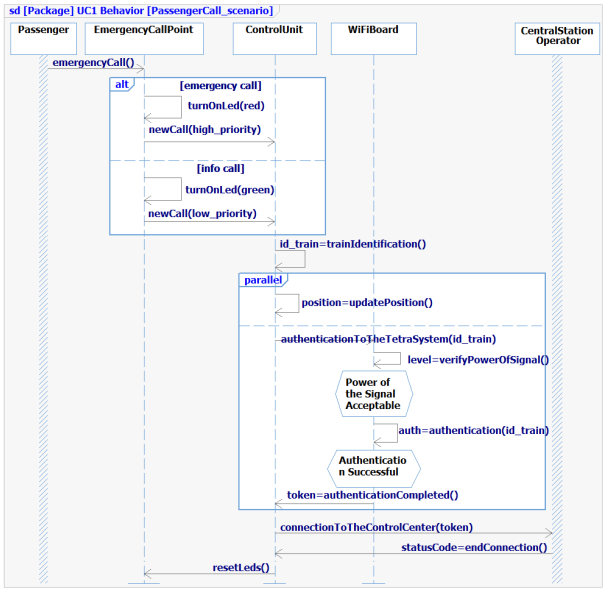

Figure 8: Intended Behavior of a Control Unit when a *Passenger Call* occurs

The modeling of both the *System Structure* and the *Intended Behavior* can be straightforward if during the system design similar structural and behavioral models have been obtained by using a UML/SysML modeling notation.

5.3. Dysfunctional Behavior Modeling

After modeling the intended behavior of systems, dysfunctional behaviors and related tasks, to represent fault and failure events and conditions, are introduced in order to analyze dependability performances through Simulation.

In particular, both the generation, management and the possible propagation of failures are modeled by considering the specific characteristics of components and then realized by combining different probability models based on popular distribution functions such as Weibull and Normal. Figures 9 and 10 show, by using SysML Activity diagrams, the tasks that represent the processes of *FaultManagement* and *FailurePropagation*.

Specifically, the *FaultManagement* task, represented in Figure 9, is able to take in input four types of fault signal: *BasePlateFault, CPUCardFault, WiFiCardFault* e *SoundCardFault*. Then, a specific activity, called *CUFaultManagement*, is in charge of managing opportunely the incoming type of fault. At the end of the management process of faults, two results can be reached: (i) the fault is handled and no other harmful consequences persist/affect in the system, (ii) the fault is not handled, so a *CUFault* signal is sent externally.

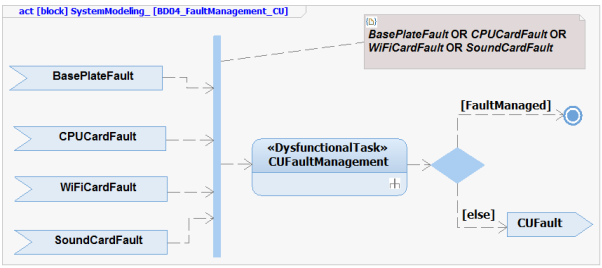

Figure 9: *FaultManagement* task of a Control Unit subsystem

The *FailurePropagation* task, represented in Figure 10, is able to take in input two types of fault signal (i.e. *Failure_24_12_DC_DC_Converter*, *CUFault*) and, after having combined and transformed them, to propagate externally a signal of failure (i.e. *CUFailure*).

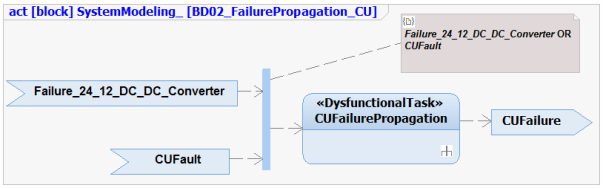

Figure 10: *FailurePropagation* task of a Control Unit subsystem

5.4. Behavior Integration

In the *Behavior Integration* activity, both the intended behaviors and the dysfunctional behaviors modeled in the previous modeling activities are integrated to obtain an overall behavioral model of the system and its component entities. As an example, in order to integrate both the *FaultManagement* and *FailurePropagation* task in the intended behavior of the *ControlUnit* subsystem, two new states have been introduced (see Figure 11) which implement the dysfunctional behavior represented in Figure 9 and Figure 10.

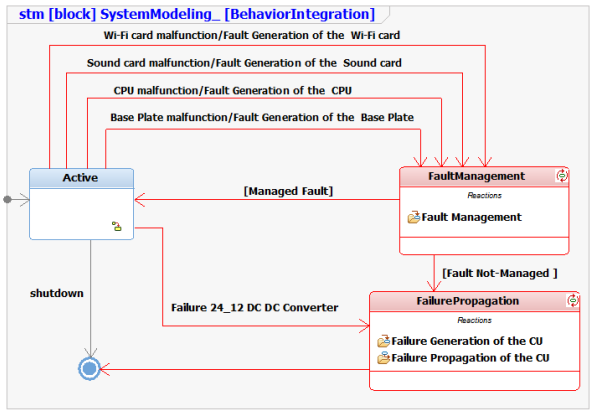

Figure 11: *BehaviorIntegration* of the Control Unit subsystem behavior

In particular, the new state machine that represents the overall behavior of the CU is now modelled by three states:

- *Active*: that performs the intended behavior of the *ControlUnit* as expected under normal operative working conditions;
- *FaultManagement:* that implements the behavior specified by the *FaultManagement* task (see Figure 9), which is responsible for managing fault signals. If the fault is handled, then the state of the *ControlUnit* comes back into the *Active* state, otherwise it changes into the *FailurePropagation* state, as described in the following;
- *FailurePropagation:* that implements both the *FailureGeneration* task for the generation of signals of faults/failures as well as the dysfunctional behavior specified by the *FailurePropagation* task of Figure 10 for the propagation of failures.

In Figure 12 the Sequence diagram shows a failure situation/scenario during the authentication process to the TETRA net when a *PassengerCall* is incoming.

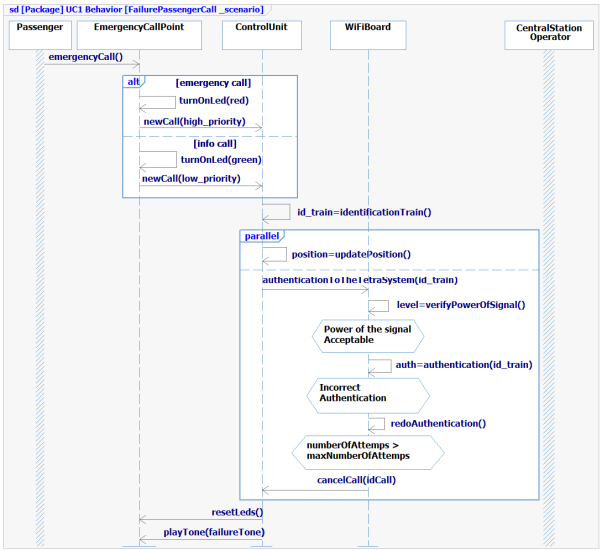

Figure 12: Failure situation of a *PassengerCall*

This activity closes the *System Modeling* phase by delivering the *System Model for Reliability Analysis* (*SMRA*) work-product.

6. SYSTEM SIMULATION

The objective of the *System Simulation* phase is to evaluate through simulation the reliability performances of the system and, possibly, compare different design alternatives and parameters setting.

The following three main activities are performed: *Model Transformation*, *Parameters Setting*, and *Simulation Execution*. Each of these activities is described in the following subsections.

6.1. Model Transformation

In this activity, the models of the system, obtained in the previous phase, are represented in terms of the constructs offered by the target simulation platform.

Indeed, a skeleton of an *Executable System Model (ESM)* is derived from the *System Models for Reliability Analysis* (*SMRA*) obtained in the *System Modeling* phase. In particular, in the current version of the RAMSAS method the *ESM* is generated for the Mathworks Simulink platform which represents a de facto standard for the simulation of multi-domain dynamic and embedded systems. This model transformation is based on a mapping between the basic SysML and Simulink constructs; in particular: (i) a (simple) SysML Block is transformed into a Simulink Block; (ii) a (composite) SysML Block, consisting of other blocks (its parts), is transformed into a Simulink Subsystem Block; (iii) SysML FlowPorts are transformed into Input and Output Simulink Blocks; (iv) SysML Flow Specifications, used to type FlowPorts, are transformed into Simulink Bus Objects. Moreover, the SysML behavioral diagrams which model the intended and the dysfunctional system behavior are transformed in Simulink functions and/or Stateflows, according to specific transformation rules.

As an example, Figure 13 sketches a Simulink model which has been derived from the On-Board Communication System represented, through a SysML notation, in Figure 2 and Figure 3.

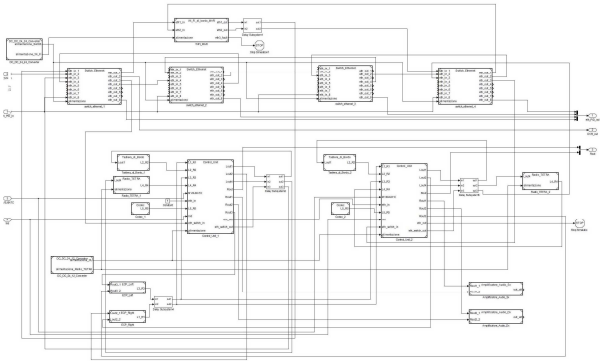

Figure 13: The Simulink model of the On-Board Communication System of the Metro train

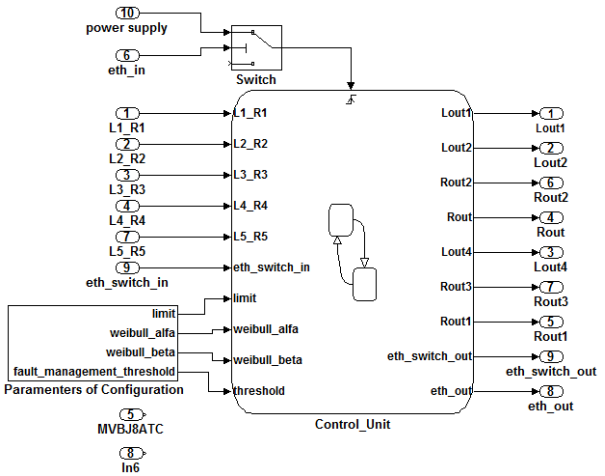

Figure 14: A High Level Simulink model of a Control Unit

As described in Section 3, the system is equipped with two *ControlUnit* (CU) that handle all the functions provided by the system. If both CU get damage, the total blockage of the system occurs. The high level representation of a CU subsystem, along with its signals, is represented in Simulink in Figure 14, which is in charge of carrying on the actions to be taken in response to an external event.

More specifically, Figure 15 shows the internal decomposition, through sub-states, of the CU behavior which in turn has been derived from the SysML diagram of Figure 11 that models the integration of the CU *Intended behavior* with its *Dysfunctional behavior*; whereas, in Figure 16 the internal behavior of the *Active* state and, in particular, of the *ManagePassengerCall*, is shown.

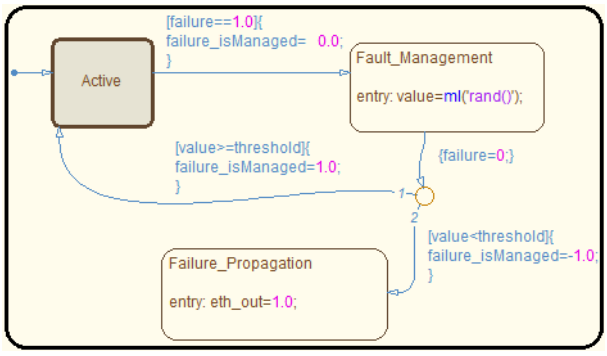

Figure 15: Internal state of a Control Unit subsystem

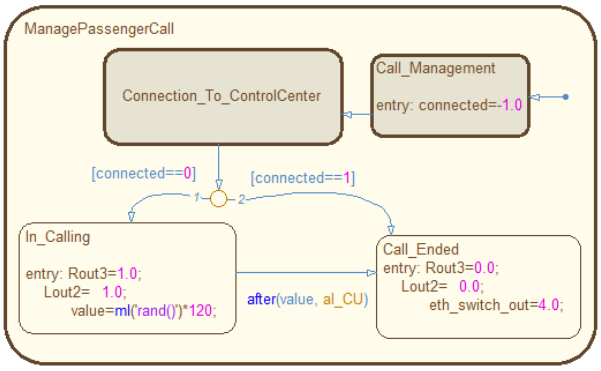

Figure 16: Stateflow of the *ManagePassengerCall* state represented in Simulink

6.2. Parameters Setting

Before starting the simulation, several system and configuration parameters can be set to evaluate system reliability performance in different simulation scenarios. In the *Parameters Setting* activity, the *ESM* is refined so to allow the flexible setting of system configuration and simulation parameters which can be tuned according to both the characteristics of the operative scenario to simulate and the failure modes to analyze (by acting on the settings of the faults and failures generation, propagation and management tasks).

In particular, each component of the system is provided of a module called “configuration parameters” in which is possible to set the parameters of the

simulation; as an example, some of them are listed in Table 2.

Table 2: Configuration Parameters with possible values

Parameter	*Description*	*Range*	*Value*
Scale Factor	It represents the statistical dispersion of the Weibull distribution	[0-∞]	1.5
Shape Factor	It defines the shape of the Weibull distribution and the position of its maximum.	[0-∞]	1
Failure Management threshold	It represents the threshold value for the management of failure	[0-∞]	0.1
Failure Generation threshold	It represents the threshold value for the generation of failure	[0-∞]	3.5
…	…	…	…

Then the model is executed according to a synchronous reactive model of computation: at each step, Simulink computes, for each block, the set of outputs as a function of the current inputs and the block state, then it updates the block state. During the simulation *faults* and *failures* are injected and/or properly caused (by *TimedEvent* or *TriggeringEvent*) in order to stress and analyze the behavior of the OBCS.

6.3. Simulation Execution and Results Assessment

The reliability indices of the Metro train system (see Section 3) have been evaluated by considering two alternatives design solutions that differ in the redundancy of the Control Unit. Indeed, the main objective was to evaluate if a second Control Unit is actually necessary to ensure the required system reliability. Indeed, a correct evaluation allows to satisfy the reliability system requirements without unnecessarily complicating the system architecture and increasing production and management costs.

In this context, in the first alternative design solution of the OBCS, the behavior of the system has been analyzed by considering a single Control Unit on board and the value of availability of the overall system has been determined according to the equation (1) reported in Section 4. In the second alternative design solution the availability of the system has been evaluated by introducing redundant elements in the overall architecture of the OBCS and, in particular, by exploiting two Control Units.

Table 3 summarizes the values of *Unavailability*, *DownTime_evMAX* and *Failure Rate (λ)* performed by the On-Board Communication System of the Metro train, respectively by employing one *CU* and two *CUs* in its architectural design.

Table 3: Performances reached respectively with One CU and Two CUs in the OBCS

	Target values	*Design with One CU*	*Design with Two CUs*
TB	Unavailability = 1-Availability = 1-0,9999 = 0,0001	Unavailability = 0,17	Unavailability = 0,0001
PB	λ= 2,31E-06 $[h^{-1}]$ *DownTime_ev MAX* = 0,16 *[h]*	λ = 4,02E-07 $[h^{-1}]$ *DownTime_ev MAX* < 0,16 *[h]*	λ = 9,01E-09 $[h^{-1}]$ *DownTime_ev MAX* < 0,16 *[h]*
Delay	λ= 4,33E-06 $[h^{-1}]$ *DownTime_ev MAX* = 0,05 *[h]*	λ=5,72E-06 $[h^{-1}]$ *DownTime_ev MAX* < 0,05 *[h]*	λ=6,33E-06 $[h^{-1}]$ *DownTime_ev MAX* < 0,05 *[h]*

Finally, the simulation results, obtained from the *Simulation Execution* phase, are analyzed in the *Results Assessment* phase with reference to the objectives of the reliability analysis identified in the initial phase of the process that are reported in Section 4.

In this case, the evaluation through simulation of the dependability performances of the system under consideration has allowed to point out some weaknesses of the OBCS design and thus to produce suggestions for its improvement before its actual realization and deployment/release. In particular, the analysis has revealed that, even though the *Failure Rate (λ)* and the *DownTime_evMAX* are quite similar when one or two Control Units are employed, the value of system reliability improves considerably with two CUs with a consequent increase in its availability. Specifically, the availability of 0,83 obtained by using one Control Unit does not satisfy the value of *availability* required (0,9999) that is easily reached by using two CUs.

7. CONCLUSIONS

Performances evaluation in railway domain and, in particular, dependability requirements analysis is very important to considerably reduce dangers to passengers travelling on the modern railways and to avoid disastrous consequences. To protect civilians against both intentional and unintentional threats, rail transportation has become increasingly automated and performance studies are fundamental to increase the lifetime of railway systems.

In this context, the paper has shown both the modeling of a real On-Board Communication System (OBCS), supplied by SELEX and installed on the Line 5 of the Milan Metro train, and the evaluation of its dependability performance by exploiting simulation techniques.

The concrete experimentation has been supported by RAMSAS, an innovative model-based method for the dependability analysis of systems. According to the RAMSAS method, the definition of a SysML-based

model, both for the intended and dysfunctional system behavior, along with the subsequent derivation of a Simulink-based simulation model have been shown.

The experimentation has led to interesting insights about the system design of the OBCS. In particular, the simulations results of the system under analysis have allowed the comparison of different design choices so as to improve the reliability and overall performances of the OBCS. Specifically, the experimentation has highlighted that the OBCS is not able to guarantee its availability if only one Control Unit is exploited, whereas two Control Units are necessary to meet the dependability requirements.

This experience has provided a further demonstration of the effectiveness of the RAMSAS method and its increasing maturity, giving useful insights for guiding its further improvements.

ACKNOWLEDGMENTS

This work has been conducted in the context of TETRis (TETRA Innovative Open Source Services), a research project funded by the Italian Ministry of Education, University and Research (MIUR) within the PON Project - Research and Competitiveness 2007/2013.

REFERENCES

ECSS-Q80-03, 2006. *Space product assurance: Methods and techniques to support the assessment of software dependability and safety*. ESA Publications Division.

European Space Agency (ESA). Available from: http://www.esa.int/.

Flammini F., 2012. Railway Safety, Reliability, and Security: Technologies and Systems Engineering. *IEEE Computer Society*, Italy.

Garro A., Groß J., Riestenpatt Gen. Richter M., Tundis A., 2013. Reliability Analysis of an Attitude Determination and Control System (ADCS) through the RAMSAS method. *Journal of Computational Science*, 5(3):439-449, Elsevier B.V., Amsterdam (The Netherlands).

Garro A., Tundis A., 2012a. Enhancing the RAMSAS method for System Reliability Analysis: an exploitation in the automotive domain. *Proceedings of the 2nd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*. July 28-31, Rome (Italy).

Garro A., Tundis A., 2012b. A model-based method for system reliability analysis. *Proceedings of the Symposium on Theory of Modeling and Simulation (TMS)*. March 26-29, Orlando (FL, USA).

Garro A., Tundis A., 2014. On the Reliability Analysis of Systems and SoS: the RAMSAS method and related extensions. *IEEE Systems Journal*. IN PRESS, IEEE Systems Council.

Garro A., Tundis A., Chirillo N., 2011. System reliability analysis: a model-based approach and a case study in the avionics industry. *Proceedings of the 3rd Air and Space International Conference (CEAS)*. October 24-28, Venice (Italy).

Guillerm R., Demmou H., Sadou N., 2010. Engineering dependability requirements for complex systems - A new information model definition. *Proceedings of the 4th Annual IEEE Systems Conference*. April 5-8, San Diego (CA, USA).

IEC-61508, 2010. Functional safety of electrical/electronic/programmable electronic safety-related systems, Parts 1-7.

International Electrotechnical Commission (IEC). Available from: http://www.iec.ch/.

ISO-26262, 2011. Software Compliance: Achieving Functional Safety in the Automotive Industry.

Krause J., Hintze E., Magnus S., Diedrich C., 2012. Model based specification, verification and test generation for a safety fieldbus profile. *Proceedings of the 31st International Conference on Computer Safety, Reliability and Security (SafeComp)*. September 25, Magdeburg (Germany).

Lahtinen J., Johansson M., Ranta J., Harju H., Nevalainen R., 2010. Comparison between IEC 60880 and IEC 61508 for certification purposes in the nuclear domain. *Proceedings of the 29th International Conference on Computer Safety, Reliability and Security (SafeComp)*. September 14-17, Vienna (Austria).

Laprie J.C., 1992. *Dependability: Basic Concepts and Terminology*, Springer-Verlag.

MATLAB-Simulink. Available from: http://www.mathworks.com/.

Molesini A., Omicini A., Ricci A. Denti E., 2005. Zooming multi-agent systems. *Proceedings of the 6th International Workshop on Agent-Oriented Software Engineering*. Utrecht (The Netherlands).

NASA, 2007. *Systems Engineering Handbook*, Revision 1, NASA/SP-2007-6105, NASA.

Navinkumar V.K., Archana R.K., 2011. Functional safety management aspects in testing of automotive safety concern systems (electronic braking system). *Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT)*. April 8-10, India.

Peraldi-Frati M., Albinet A., 2010. Requirement traceability in safety critical systems. *Proceedings of the 1st Workshop on Critical Automotive applications: Robustness & Safety (CARS)*. April 27, Valencia (Spain).

Reliability and Safety in Railway, 2012. Edited by Xavier Perpinya, pp. 418.

Rierson L., 2013. *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance*, CRC Press.

RTCA/DO 254, 2000. *Design Assurance Guidance For Airbone Electronic Hardware*.

Stapelberg R.F., 2008. *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*, 1st ed. Spinger-Verlag.

Scott A., Dadashi N., Wilson J. R., Mills A., 2013. *Rail Human Factors: Supporting reliability, safety and cost reduction*. Taylor & Francis.
SELEX ES – A Finmeccanica Company. Available from: http://www.selex-es.com/.
Systems Modeling Language (SysML). Available from: http://www.omgsysml.org/.
TETRA Innovative Open Source Services (TETRis). Available from: http://www.ponrec.it/open-data/progetti/scheda-progetto?ProgettoID=5013.
Tundis A., Rogovchenko-Buffoni L., Fritzson P., Garro A., 2013. Modeling System Requirements in Modelica: Definition and Comparison of Candidate Approaches. *Proceedings of the 5th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools (EOOLT 2013)*. April 19, Nottingham (UK).
Yu G., Xu Z., Du J., 2009. An Approach for Automated Safety Testing of Safety-Critical Software System Based on Safety Requirements. *Proceedings in the International Forum on Information Technology and Applications (IFITA)*. May 15-17, Chengdu (China).

AUTHOR BIOGRAPHIES

Alberto Falcone

Alberto Falcone received the Laurea Degree in Computer Engineering from the University of Calabria (Italy) in 2011 and a Master title in Industrial Research from the same institution in 2013. He is a PhD student in Information and Communication Engineering for Pervasive Intelligent Environments at Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES) at University of Calabria. His main research interests include Modeling and Simulation of Systems. He has been involved as a Team Leader in the Simulation Exploration Experience (SEE) 2014, a project lead by NASA, in which he has contributed to the design and development of a module able to provide surface-to-surface communication services among the entities that populate a Moon base.

Alfredo Garro

Alfredo Garro is an Associate Professor of Computing Systems at the Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES) of the University of Calabria (Italy). From 1999 to 2001, he has been a researcher at CSELT, the Telecom Italia Group R&D Lab. From 2001 to 2003, he collaborates with the Institute of High Performance Computing and Networking of the Italian National Research Council (CNR). On February 2005 he received the PhD Degree in Systems and Computer Engineering from the University of Calabria. From January 2005 to December 2011, he has been an Assistant Professor of Computing Systems at the DIMES Department (formerly DEIS) of the University of Calabria. His main research interests include: systems and software engineering, reliability engineering, modeling and simulation. His list of publications contains about 80 papers published in international journals, books and proceedings of international and national conferences. He is a member of the IEEE and IEEE Computer Society from 2005. He is a member of the IEEE Reliability Society and IEEE Aerospace and Electronic Systems Society. Currently, he operates as a member of the SPACE Forum Planning and Review Panel (PRP) of the Simulation Interoperability Standards Organization (SISO). He is member of the Executive Committee of the MODRIO (Model Driven Physical Systems Operation) ITEA2 Project and the Technical Contact for his Institution in the Open Source Modelica Consortium (OSMC).

Andrea Tundis

Andrea Tundis received the Laurea Degree in Computer Engineering from the University of Calabria (Italy) in 2009, a Master title in Industrial Research from the same Institution in 2010, and, on February 2014, a PhD Degree in Systems and Computer Engineering from the University of Calabria where he is currently a research fellow. His main research interests include the definition of model-based methods for the reliability and safety analysis of systems as well as models for the formalization and traceability of non-functional requirements. In the last year, he worked at the Programming Environment Laboratory (PELAB) at Linköping University (Sweden) on the extension of the Modelica language for the modeling of system properties.