System dependability assessment -Interplay between research and practice

Mohamed Kaâniche, Karama Kanoun

System dependability assessment - Interplay between research and practice

Mohamed Kaâniche and Karama Kanoun

LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
7 avenue du colonel Roche, F-31400 Toulouse, France

Abstract

This chapter illustrates examples of collaborative work with industry in which we have been involved over time. We concentrate on a set of projects one or both authors contributed to, in the area of dependability assessment. These collaborations are grouped into four main topics, corresponding respectively to: model-based dependability assessment, software reliability, simulation and fault injection based dependability assessment, online error detection and diagnosis. We show examples of results obtained in the framework of these collaborations.

1 Introduction

This chapter aims to briefly present examples of collaborative work with industry in which we have been involved over time. We concentrate on a set of projects one or both authors contributed to, in the area of dependability assessment. Assessment is used in a broad sense, including dependability evaluation based on i) behavioral system models, or on ii) data collected on a real system, or on iii) controlled experiments, mainly based on fault injections. The assessment has been used either at design time to help define system architecture and components, or at run time, to characterize potential system misbehavior or to help and plan corrective or maintenance actions.

The topics addressed in the collaborative work evolved during the years, initially, taking into account mainly hardware accidental faults and concentrating on system safety, then integrating progressively software design faults, either considered alone, or considering both hardware and software faults and their interactions. Later, we have focused on the assessment of the impact of malicious faults on system security, considered alone, then jointly addressing safety and security together. More recently, model learning approaches used at run time, for error detection and diagnosis, have become central.

The context in which our collaborations with industry have been triggered can be classified into two extreme situations:

- The company has immediate needs, together with a very specific problem to solve in short or near terms.
- The company needs are real, but the problem to address is open, it is not clearly stated (on purpose), this case usually corresponds to advance long-term prospection, to increase the company's technical skills and be ready for new challenges.

Of course, intermediate situations are not uncommon: starting with a very specific problem to be immediately solved, the ultimate aim of the work could be to go further and prepare new changes for the company.

Obviously, long-term objective situations, without very specific problems to solve a priori, are more challenging for research, as they are usually far-reaching and ambitious. The topics are then defined jointly with the company. On the other hand, very specific problems, requiring immediate solutions, might be less challenging for research, in particular when similar problems have already been solved in other sectors. In this case, research role could be to open the company's vision to leading practices in other sectors, and can go further by generalizing the problem and providing solutions that are as generic as possible.

Indeed, the company is a priori interested in solutions that correspond to their specific business, and research is interested in real problems to solve that could fit other sectors.

In the rest of this chapter, we will briefly present examples of work we have carried out in collaboration with industry. These examples are grouped in four sections: i) model-based dependability assessment, ii) software reliability, iii) simulation and fault injection based dependability assessment, iv) online error detection and diagnosis

2 Model-based dependability assessment

Usually, dependability assessment is carried at design time, in order to define an appropriate system architecture satisfying the system dependability requirements. However, model-based dependability assessment could also be helpful during system operation to anticipate system failures and to schedule system maintenance and/or reschedule system mission accordingly. Our work, in an industrial context, addressed essentially the design phase, with an exception, related to aeronautics. Even though the same modeling techniques can be used at runtime, the added constraints concern essentially the way models should be structured to allow their quick update (and re-execution) during system operation, without requiring an additional validation.

The main difficulties when modeling real-life systems come from the inherent complexity of the system performing a multitude of related or correlated functions, which induces large-scale models, requiring large processing times and tedious validation process. In addition, during system design, assessment is used as a powerful means for comparing possible system architectures to select the most suitable one. Modeling approaches are thus needed to optimize model construction of competing architectures, their processing and validation, to encourage comparison of several of them. This is also true for assessment at runtime, as one has to compare maintenance strategies that are possible, depending on the nature of failed components and on the maintenance environment available at the time of failure.

Defining an appropriate dependability measure (or attribute) to be assessed is fundamental. Indeed, classical measures such as availability of safety do not necessarily allow bringing out the most salient properties of the system. In addition, several complementary attributes are most of the time required to highlight various facets of a system.

Our work related to dependability assessment was based on Markov chains and evolved towards Generalized Stochastic Petri Nets (GSPNs) and their offspring. We have also explored how to automate the generation of GSPN based model from higher levels description languages such as AADL (Architecture Analysis and Design Language), a mature industry-standard, (Rugina et al. 2011). To illustrate the kind of modeling activities we have carried out in collaboration with industry, we have selected three critical domains: electricity production and distribution, air traffic control, and aeronautics.

Electricity production and distribution

Several collaborations took place with EDF, the French multinational electric utility company. The first one (Laprie and Kanoun 1980) was dedicated to the definition of a new architecture for the monitoring systems of an Extra High Voltage substation, driven by rare, external solicitations, due to incidents in the process. The main difficulties were due to i) the dormancy of the considered system, some parts of which could be failed for a long period of time and could not be detected without the occurrence of an incident or until the next system inspection, and ii) there were very few model processing tools well suited to this kind of systems and we have adapted SURF-2 (Béounes et al. 1993) to this end. The approach we have followed was based on i) the definition of dependability levels and attributes, directly from the statement of system functions, that can be physically interpreted, and ii) the construction of the systems dependability model by aggregating models of subsets that have been reduced to include only those pa-

rameters that have a real influence on the considered attribute. The results have shown the impact of redundancy both at the computing system level (using double or triple redundancy for all local equipment) and at the communication level (a reconfigurable optical counter-rotating double loop) (Blanquart et al. 1982). Even though the above results addressed only the hardware part of the systems and seem obvious nowadays, the work performed was challenging because of its explorative nature at that time.

A more recent collaboration with EDF was dedicated to the selection of Instrumentation and control system based on modeling together with fault injection. Candidate architectures proposed by various suppliers were compared based on GSPNs (Betous-Almeida and Kanoun 2004). The most impacting parameters identified are then evaluated experimentally using fault injection. (Betous-Almeida et al. 2000).

Air Traffic control

As the air traffic control (ATC) volume is continuously and rapidly growing, the associated control systems have to evolve to meet this trend. The French ATC is based on an automated system (the CAUTRA, "*Coordinateur Automatisé du Trafic Aérien"*) providing a valuable support to controllers. CAUTRA is implemented on a distributed fault-tolerant computing system composed of five regional control centers (RCC) ensuring coverage of the whole country, and a centralized operating center, connected to these centers through a dedicated telecommunication network. Redundancy and fault tolerance mechanisms are used at various hardware and software component levels.

In the framework of two joint collaborations, we have developed two modeling approaches, dedicated respectively to the assessment of:

- RCC availability: with respect to its two main functions, Flight Plan Processing (FP) and Radar Data Processing (RD), based on the analysis of the impact of the failures of its own components, (see e.g. (Kanoun et al. 1999)).
- ATC safety: based on the analysis of the impact of the CAUTRA components' failures on the degradation of the service provided to the controller, including the global CAUTRA system architecture with the five interconnected RCC subsystems interconected (Fota et al. 1999).

For each case, we have defined and modeled several alternative architectures, compared their dependability measures, and identified the most important factors impacting these measures. Even though the two modeling approaches are i) modular, ii) based on GSPNs and iii) take into account permanent and transient failures of hardware and software components as well as error propagation between components, they differ in the model construction approaches due to the very different nature of the final measures to be assessed. Availability is defined at the regional

center level, while safety is evaluated at the CAUTRA level requiring the knowledge of the states of all centers. Hence, in addition to modeling of the five regional centers and the centralized operating center, ATC safety analyses rely heavily on a detailed Failure Modes and Criticality Analysis (FMECA) to define and assess the service degradation levels.

For example, for RCC availability assessment, a block modeling approach has been defined to assess the impact of reconfiguration strategies on the availabilities of FP and RD. A block represents the model of a component or a dependency. The block models are generic and have well-defined interfacing rules to facilitate their composition as well as validation of the resulting composed models. Hence, several reconfiguration strategies have been easily compared, with only few additional blocks, compared to the current architecture at that time. The comparative analyses identified the reconfiguration strategies that satisfy the following important requirement: RD unavailability should be less than 5 min per year.

For ATC safety, the models of the regional and the centralized centers are built independently in successive steps according to an incremental approach, following specific construction guidelines, to assess the centers dependability, together with an appropriate specification language associated with transformation rules allowing an automatic generation of optimized GSPNs. The resulting models are very complex. For example the GSPN of the Radar Data Processing and the Flight Plan Data Processing Systems has about 100 places and 500 transitions and corresponds to a reduced Markov chain of about 25 000 states. At the global level, the partial measures assessed for the six centers are combined to assess their impact on ATC safety, more precisely, on the levels of degradation of the service provided to the controller. The results identified the most impacting features that need to be monitored during the design phase.

Aeronautics

As stated earlier, model-based dependability assessment can be helpful during system operation too, to anticipate system failures (or mission interruption). This is typically the case in aeronautics where airline companies and operators need to reschedule a mission if some components fail. Our collaboration with Airbus was in this context.

The main challenge comes from the fact that the model has to be tuned dynamically, in operation, to take into account the current state of the system, the maintenance environments and potential new information by operators. Indeed, these operators usually do not have any knowledge related to dependability modeling techniques. Hence the model should be prepared and validated in advance, offline, in a way that makes it easily and very quickly configurable in operation.

To this end, we have developed a modeling approach, based on a meta- model used to i) structure the information needed to assess operational reliability, and to ii) build a stochastic model to be tuned dynamically to take into account the system operational state, the mission profile and the maintenance facilities (Tiassou et al 2012). This model allows to i) assess, on-the-fly, the ability to succeed in continuing on the remaining part of the mission, in case of an unscheduled event occurrence, and to ii) support maintenance planning. A case study, based on an aircraft subsystem, is considered for illustration, using the Stochastic Activity Networks formalism. It shows how to re-schedule a mission, based on the failed component and its impact on the remaining part of the mission, as well as the maintenance possibilities at the various stops of the aircraft (Tiassou et al. 2013).

3 Software reliability

Even though the first software reliability growth models have been published in early 70s, their practical use in industrial environments was very seldom at the time we had our collaborations with industry. Our first investigations showed that preliminary rigorous analyses of failure data are required before model applications. For example, reliability growth/decrease tests i) tell about the impact of fault removal on software reliability evolution and ii) guide selection of models to be applied according to their trend. In particular, reliability growth model assuming pure reliability growth give non-meaningful results if applied to data displaying reliability decrease over some periods of time. Data partitioning improves model results.

We briefly report on two collaborations concerning Electronic Switching Systems (ESSs), developed by two different companies Alcatel and TELEBRAS. Even though both companies were interested in assessing software behavior in operation, the measures of interest were different, due essentially to the different maturity levels of the software, when the study took place.

For Alcatel, the dataset analyzed was collected during 3 years, on a mature system, in operation on more than 1000 sites. One of the expected results of the collaboration was an estimate of the software failure rate. Alcatel aim was to build a Markov chain, taking into account hardware and software failures, to assess the switching system unavailability, to check its compliance with the international telecommunications requirements (that was less than 3 min / year).

Detailed analyses of the data either at the level of the ESS, at the components levels, as well as taking into account the severity of failures (impact of failures on service loss) is performed in (Kanoun and Sabourin 1987). They show for example that: i) the defense component (in charge of hardware fault tolerance) has the highest failure rate, the three other software components dedicated to functional

operation have equivalent, lower, failure rates, ii) only a very low number of faults led to service unavailability, the others had minor impact.

TELEBRAS started the development of a new series of ESSs increasing progressively the ESS capacity. We concentrated first on the software of the early one before considering together three successive generations. This very first analysis (Kanoun et al. 1991) allowed to gain insight into the development process and provided, among other things, an estimate of the number of failures that will occur in the field (equivalently, the number of required corrections), for planning the maintenance effort after system delivery. A comparative analysis of the reliability, in terms of failure rates, of the three generations gives insight into the evolution of the reliability of a family of products (Kaâniche and Kanoun1994). Examples of comparative analyses, with respect to the evolution of the nature of faults activated and their impact through the successive generations, can be found in (Kaâniche and Kanoun1998).

4 Dependability assessment based on simulation and fault injection

Simulation offers complementary means to analytical modeling for analyzing and assessing dependability. Specifically, it offers the possibility to take into account a much wider spectrum of assumptions and to describe the system behavior at a relatively lower level of detail, in order to analyze the effects of faults as close as possible to the components where they occur, and to study their impacts at the system level. Also, it can be used to estimate the parameters involved in analytical dependability models. The main challenge is related to the need to master the simulation time that increases dramatically when the model is simulated at a low level of detail. One possible solution is to develop a hierarchical simulation approach to analyze the behavior of the target system in the presence of faults by considering different levels of abstraction. We have explored this approach in the context of a collaborative research project involving the University of Illinois at Urbana-Champaign, and the StorageTek Company in Colorado, USA [Kaaniche et al. 1998]. This approach was developed to support the dependability analysis and evaluation of a highly available commercial cache-based RAID storage system. The architecture is complex and includes several layers of overlapping error detection and recovery mechanisms. Three abstraction levels have been considered to model the cache architecture, cache operations, and error detection and recovery mechanism. The impact of faults and errors occurring in the cache and in the disks was analyzed at each level of the hierarchy. The models have been developed using the DEPEND simulation-based environment developed at UIUC, which provides facilities to inject faults into a functional behavior model, to simulate error detection and recovery mechanisms, and to evaluate quantitative measures. Several fault models were defined for each submodel to simulate cache component

failures, disk failures, transmission errors, and data errors in the cache memory and in the disks. Some of the parameters characterizing fault injection in a given submodel correspond to probabilities evaluated from the simulation of the lower-level submodel. Based on the proposed methodology, we evaluated and analyzed i) the system behavior under a real workload and high error rate (focusing on error bursts), ii) the coverage of the error detection mechanisms implemented in the system and the error latency distributions, and iii) the accumulation of errors in the cache and in the disks. It is important to emphasize that an analytical modeling of the system is not appropriate in this context due to the complexity of the architecture, the overlapping of error detection and recovery mechanisms, and the necessity of capturing the latent errors in the cache and the disks.

Another major challenge in industry concerns the efficient integration of dependability assessment techniques into their existing system engineering process and tools. We had the opportunity to explore this challenge with Technicatome, a French industrial leader in nuclear engineering and with Valeo, a global worldwide automotive supplier.

With Technicatome, the objective was to rely on a commercial systems engineering tool (RDD-100) in order to facilitate the integration of operational safety analyses in industrial processes at early design stages. This work has resulted in the extension of the functionalities offered by this tool by defining mechanisms for injecting faults into RDD-100 models and analyzing their effects from the point of view of operational safety. The proposed mechanisms are based on two complementary techniques. The first one consists in inserting saboteurs, to disturb the inputs or outputs of the elementary components of the nominal model as well as their behavior. The second one consists in directly mutating the code of the elementary components of the nominal model. A critical analysis of the advantages and limitations of each of these techniques, in terms of difficulty of implementation and the possibilities offered for fault injection and observation of their effects, led us to propose a solution combining these two techniques [Kaâniche et al 2004].

Collaboration with Valeo took place in the context of the publication of the first standard specifically dedicated to automotive safety systems, ISO 26262. This standard requires introducing fault injection from the very early phases of the development process. Indeed, even though experimental validation of embedded systems, including fault injection, was of common practice in industry, its adoption in the early design phase, as advocated by the ISO26262 standard, was not common and unclear. In this context, we developed a global approach integrating fault injection in the whole development process in a continuous way, from system requirements to the verification and validation phase. We have shown the strong link between classical safety analyses, commonly used in industry for critical systems design and fault injection principles at design phase. In particular, we have shown the similarities between two well-known domains that are separated in practice, namely i) Failure Modes, Effects, and Criticality analyses Analysis (FMECA) and ii) fault injection. More precisely, we have shown how FMECA

spreadsheets i) can be used to guide fault injection on one hand, and to synthesize the results of fault injection in the other hand, and ii) to link the successive development levels via their failure modes, their causes and their effects, to capture the failure propagation paths between the levels. These chains help making fault injection campaigns effective.

We have shown the benefits of the proposed approach (Pintard et al. 2014), which is compliant with the ISO 26262 standard, on a case study from the automotive domain. The ultimate aim was to guide fault injection experiments, based on early system safety analyses to optimize the whole development process by defining an optimal set of experiments. From a practical point of view, a fault injection tool was developed by Valeo to implement fault injection at various levels.

5 Online error detection and diagnosis

Traditional approaches to dependability assessment are based on the development of models during the design phase in order to assist in architectural choices. However, the need is more and more for automated solutions allowing to monitor and assess the dependability of the system at run-time, i.e. while the system is in operation, using in particular machine learning algorithms. Indeed, the massive collection of data together with the significant progress and successes achieved with machine learning algorithms have motivated the exploration of these algorithms to support anomaly detection and diagnosis in several application areas. It should be noted that such models have been studied since the 1980s, but have received increased attention in the recent years due to the growing interest in artificial intelligence techniques, particularly in industry.

We have explored the use of such techniques to support online error detection and diagnosis in cloud infrastructures and future telecommunication architectures using network functions and software virtualization technologies (SDN and NFV). This study was carried out in collaboration with Orange Labs. In particular, we have defined a generic strategy enabling the detection of two types of anomalies in cloud services (errors and service level agreement violations) while providing two diagnosis levels to the cloud provider (i.e., identifying the anomalous virtual machine and the type of error causing the anomaly) (Sauvanaud et al. 2018). The strategy is based on system monitoring data collected online either from the monitored cloud service, or from the underlying hypervisor(s) hosting the service. Different types of machine learning algorithms (supervised, unsupervised, and hybrid) were used to classify anomalous behaviors of the service. Moreover a fault injection tool was developed to collect training data including anomalous samples to train the detection and diagnosis models and to validate our detection strategy. The evaluation was applied to two case studies: a database management system (MongoDB) and an IP multimedia system developed as a virtual network function.

In particular, we have compared the efficiency of several classification algorithms and concluded that the Random Forests algorithm provides in our context the best tradeoff in terms of detection efficiency and training and detection time. The experimental results include a comparative analysis of the detection performance obtained with Operating Systems related monitoring data and hypervisor monitoring data.

We have explored similar approaches in the security area to support the detection of potential intrusions targeting critical embedded applications. In particular, in the context of a joint work with Thales avionics, we designed and implemented a host-based intrusion detection system (HIDS) adapted to the specific constraints and stringent requirements of real-time critical applications embedded in Integrated Modular Avionics (IMA) architectures [Damien et al. 2019-b]. The proposed HIDS implements an anomaly-based approach based on the monitoring of ARINC 653 API calls. The model of the legitimate behavior of the application is built based on data collected during the aircraft integration phase. Besides detecting anomalous behaviors of the target application, a signature-based system providing a first diagnosis after the detection of an anomalous behavior was also implemented. This approach has been validated on a real avionic computer and yielded good results in terms of classification accuracy and resource consumption [Damien et al. 2020]. To support the validation of the HIDS, we developed a tool enabling the automatic injection of attacks and the generation of application code mutations that mimic the behavior of malevolent pieces of code introduced inside the target application [Damien et al. 2019-a].

Besides avionics, we also had the opportunity to design and implement, in collaboration with Renault, an intrusion detection system for CAN (Controller Area Network) based embedded automotive networks, that takes into account specific constraints of the automotive domain (simplicity of implementation without modifying the ECU (Electronic Control Unit) architecture, low cost, low detection latency). The proposed approach consists in automatically generating attack signatures from automata based models, derived from the ECU specification, describing the behavior of the ECUs on board interacting via messages on the CAN bus (Studnia et al 2018). This approach was validated on an early prototype using simulated attacks performed on logs of an actual CAN network.

6 Lessons learned and concluding remarks

Joint collaborations with industry have always been an important source of inspiration for new research challenges and solutions, as well as an opportunity to validate our results on real-life use cases and applications. Such collaborations have allowed us to address a wide range of topics with several industry partners

from various application domains (aeronautics, automotive, telecommunication, energy, etc). Development of scalable and generic solutions that can accommodate the increasing complexity of computing systems and be easily integrated within industrial system engineering processes are concerns shared by several industry partners. The examples presented in this chapter clearly show the evolution of the topics of interest to industry over the years, at least as we have experienced it from our side. This evolution is in-line with the evolution of the technological trends through the years.

Over the years, we learned to reach quickly a consensus between industry needs and research objectives while preserving intellectual and industrial properties. In particular, we have accepted not to publish all results. In an academic world where publication is becoming more and more a driving process, this may be not acceptable. However, we have always been able to agree on i) parts of results we were allowed to publish and/or include in the PhD dissertation without harming the industrial properties and ii) results to be delivered to the industrialist only. With the benefits of hindsight, this process was always challenging and extremely rewarding. Another lesson learned concerns the clear distinction, upfront, between the user and the system provider perspectives. This led us to define dependability attributes related to the two points of view, and try to optimize both of them or at least reach an acceptable compromise, which was not obvious all the time, but truly challenging too.

Currently, significant efforts in many industries are being devoted to exploring the opportunities opened by recent advances in artificial intelligent techniques together with massive data collection thanks to the widespread deployment of IoT technologies. One of the main questions is how to leverage these techniques in various areas (monitoring, anomaly and intrusion detection and diagnosis, testing, etc.) to improve the dependability, resilience and quality of service of networks and computer systems performance while reducing the costs. A major challenge in this context is related to the lack of trust and confidence in such techniques and the need for rigorous and formalized approaches to provide justified assurance and ensure a better explainability and acceptability of AI algorithms in a context where malicious threats are also increasing. This problem has many dimensions and requires an interdisciplinary approach combining expertise from various scientific fields including mathematics, optimization theory, computer science, and also social and neuro-sciences. The study of use cases in different application domains and the cross-fertilization of solutions from different industrial sectors will be a key to achieve significant advances in this field.

Acknowledgement

The chapter topic was inspired by friendly discussions with Ravi Iyer, over the years, especially during our sabbatical visits to the University of Illinois at Urbana Champaign or at LAAS-CNRS in Toulouse. We all know that Ravi enjoys the practical aspects of research, in addition to deep conceptual work. We had the opportunity to share with him our lessons learned and some ideas inspired from the industry projects that we have conducted as well as from our experimental-oriented research on system dependability and software reliability, and also more recently on the assessment and mitigation of security threats. Thank you Ravi!

We also thank all our numerous colleagues from LAAS-CNRS and from industry with whom we have carried out the collaborative work outlined in this chapter.

References

(Béounes et al. 1993): C. Béounes, M. Aguera J. Aalat, S. Bachmann, C. Bourdeau, J.E. Doucet, K. Kanoun, J.C. Laprie, S. Metge, J. Moeira De Souza, D. Powell, P. Spiesser. SURF-2: a program for dependability evaluation of complex hardware and software systems. 23rd IEEE International Symposium on Fault-Tolerant Computing (FTCS'23), Toulouse (France), 22-24 June 1993, pp.668-673

(Betous-Almeida and Kanoun 2004): C. Betous-Almeida, K. Kanoun. Dependability modelling of instrumentation and control systems: a compaaprison of competing architectures. Safety Science, Vol.42, N°5, pp.457-480, June 2004

(Betous-Almeida et al. 2000): C. Betous-Almeida, A. Arazo, Y. Crouzet, K. Kanoun. Dependability of computer control systems in power plants. Analytical and experimental evaluation. 19th International Conference on Computer Safety, Reliability and Security (SAFECOMP'2000), Rotterdam, 24-27 October 2000.

(Blanquart et al. 1982): J.P. Blanquart, J.L. Boussin, K.Kanoun, J.C. Laprie, REBECCA: a dependable communication sub-system for the control system of extra-high-voltage substations. Fifth European Conference on Electrotechnics, Lyngby (Danemark), 14-18 June 1982, pp. 825-829.

(Damien et al 2019-a) A. Damien, N. Feyt, V. Nicomette, E. Alata, M. Kaâniche, Attack Injection into Avionic Systems through Application Code Mutation, 38th Digital Avionics Systems Conference (DASC-2019), 08-12 September 2019, San Diego, CA (US), 2019, 10p.

(Damien et al 2019-b) A. Damien, M. Marcourt (1), V. Nicomette, E. Alata, M. Kaâniche, Implementation of a Host-based Intrusion Detection Systems for Avionic Applications, 24th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC-2019), Kyoto, 1-3 December 2019, IEEE CS, IEEE CS, 10p

(Damien et al 2020) A. Damien, P.-F. Gimenez, N. Feyt, V. Nicomette, M. Kaâniche, E. Alata. On-board Diagnosis: A First Step from Detection to Prevention of Intrusions on Avionics Applications. 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE-2020), October 2020, Coimbra, Portugal, pp.358-368.

(Fota et al. 1999): N. Fota, M. Kaâniche, K. Kanoun, Dependability evaluation of an air traffic control computing system, Performance Evaluation, Vol. 35, N°3-4, pp. 253-273, May 1999.

(Kaâniche and Kanoun, 1998): M. Kaâniche, K. Kanoun, Software reliabilty analysis of three successive generations of a telecommunications system, IEEE Workshop on Application-Specific Software Engineering and Technology (ASSET'98), Richardson (USA), March 26-28 1998, pp. 122-127

(Kaâniche et al. 1998): M. Kaâniche, L. Romano, Z. Kalbarczyk, R. Iyer, R. Karcich, A hierarchical approach for dependability analysis of a commercial cache-based RAID storage architecture, 28th IEEE International Symposium on Fault-Tolerant Computing (FTCS-28), Munich (Germany), June 1998, pp.6-15

(Kaâniche et al. 1994): M. Kaâniche, K. Kanoun, M. Cukier, and M. Bastos Martini, Software Reliability Analysis of Three Successive Generations of a Switching System. First European Conference on Dependable Computing (EDCC-1), Berlin, Germany, 4-6 October 1994, pp. 473-490.

(Kaâniche et al. 2004) M. Kaâniche, Y. Le Guédart, J. Arlat, T. Boyer, An investigation on mutation strategies for fault injection into RDD-100 models, Safety Science, Issue 5, Vol.42, pp.385-403, June 2004

(Kanoun et al. 1991): K. Kanoun, M. Bastos Martini, and J. Moreira de Souza. A Method for Software Reliability Analysis and Prediction— Application to The TROPICO-R Switching System. IEEE Transactions on Software Engineering, vol. 17, pp. 334-344, 1991.

(Kanoun and Sabourin, 1987): K. Kanoun, T. Sabourin Software dependability of a telephone switching system. 17th IEEE International Symposium on Fault Tolerant Computing (FTCS-17), Pittsburgh (USA), 6-8 July 1987, pp.236-24.

(Kanoun et al. 1999): K. Kanoun, M. Borrel, T. Morteveille, A. Peytavin, Availability of CAUTRA, a subset of the French air traffic control system, IEEE Transactions on Computers, Vol.48, N°5, pp.528-535, May1999

(Laprie and Kanoun, 1980): J.C. Laprie and K. Kanoun. Dependability Modeling of Safety Systems, 10th IEEE International Symposium on Fault-Tolerant Computing, 1-3 October 1980, pp. 245-250.

(Pintard et al. 2014): L. Pintard, J.C. Fabre, K. Kanoun, M. Leeman, M. Roy, From Safety Analysis to Experimental Validation of Automotive Systems", IEEE Pacific Rim Dependable Computing Conference (PRDC 2014), 19–21 November. 2014, Singapore.

(Rugina et al. 2011): A. E. Rugina, K. Kanoun, M. Kaâniche. Software Dependability Modeling Using AADL (Architecture Analysis and Design Language). International Journal of Performability Engineering, Vol.7, N°4, pp. 313-325, July 2011.

(Sauvanaud et al. 2018) C. Sauvanaud, M. Kaâniche, K. Kanoun, K. Lazri, G. Da Silva Silvestre. Anomaly Detection and Diagnosis for Cloud services: Practical experiments and lessons learned, Journal of Systems & Software, Special issue on Software Reliability, Volume 139, May 2018, Pages 84-106.

(Studnia et al 2018): I. Studnia, E. Alata, V. Nicomette, M. Kaâniche, Y. Laarouchi, A language-based intrusion detection approach for automotive embedded networks, International Journal on Embedded Systems. Volume 10, Issue1, 2018.

(Tiassou et al. 2012): K. Tiassou, K. Kanoun, M. Kaâniche, C. Seguin, C. Papadopoulos. Impact of Operational Reliability re-Assessment during Aircraft Missions. 31st IEEE International Symposium on Reliable Distributed Systems (SRDS 2012), Irvine, CA, USA, 8-11 October 2012, pp. 219-224.

(Tiassou et al. 2013): K. Tiassou, K. Kanoun, M. Kaâniche, C. Seguin, C. Papadopoulos. Aircraft operational reliability — A model-based approach and a case study. RESS (Reliability Engineering and System Safety), 120 (2013) pp. 163–176.