Available online at www.sciencedirect.com
SCIENCE DIRECT®
Reliability Engineering and System Safety 81 (2003) 235–238
RELIABILITY ENGINEERING & SYSTEM SAFETY
www.elsevier.com/locate/ress

Guest Editorial

Safety, reliability and security of industrial computer systems

This special issue consists of the best papers presented at SAFECOMP 2002 [1] held in Catania, Italy, 10–13 September 2002. Since its establishment in 1979 by the European Workshop on Industrial Computer Systems[1] (EWICS), SAFECOMP, the series of conferences on Computer Safety, Reliability and Security[2], has contributed to the progress of the state of the art in dependable applications of computer systems. SAFECOMP provides ample opportunity to exchange insights and experiences in emerging methods and practices across the borders of different disciplines. Previous SAFECOMPs have already registered the increasing interest for multi-disciplinary approaches dealing with system dependability in previous proceedings[3] and journal special issues [2,3]. SAFECOMP 2002 further addresses multi-disciplinarity by stressing the collaborations of different scientific communities towards the identification of communication means that may enhance our ability in designing, building, deploying and assessing computer systems.

The recent developments in system dependability emphasize a shift from process-based development towards model-based development from early stage to certification, deployment, evaluation, evolution, and decommission in the system life cycle. There are two extreme strategies in order to investigate model-oriented system dependability. On the one hand we might be tempted to looking for a universal model that may be used throughout the entire life cycle. A universal model should support not only different stages in the system life cycle, but should also support different stakeholders who drive the system life cycle as well as the dependability system requirements. Therefore a universal model may ever remain a utopia in systems engineering. On the other hand we face the practical challenge of integrating different existing models. The emergence of ubiquitous and pervasive technologies emphasizes the socio-technical nature of computer systems, which were classically treated in isolation from their surroundings. From an historical viewpoint the stone age of computer systems has just passed. There is now a common understanding that computer systems are not just a combination of hardware and software, but they involve human beings as well. Thus the design of computer systems requires a subtle understanding of human and computer behaviors.

We are now seeking methodologies that support a holistic approach to the design of dependable computer systems. A holistic viewpoint emphasizes how computer systems are ubiquitous and pervasive in the modern electronically mediated society commonly known as the Information Society. Citizens of the electronically mediated society are socio-technical systems contributing to emergent behaviors of the whole. They require an incremental approach to design that guides them (step-by-step) through their evolution from the early to the mature age. Different levels of acceptance and diverse granularity characterize their life cycle. In order to support socio-technical systems we need further to investigate how to integrate different methodologies throughout the entire life cycle. The integration of different methodologies requires us further to understand how to embed formalities (you might prefer the term formal methods) in the system life cycle as well as how to take account of human factors in engineering dependable systems. These challenges tell us that the Renaissance of the electronically mediated society is still far in the future. This special issue represents a small step forward towards system dependability. The papers in this special issue draw directions than blend practical experience and research studies in engineering dependable systems.

Safety as well as dependability is a property of the whole system. Each component contributes to the overall safety. Different techniques can tackle safety analysis. The integration of different methodologies enhances our understanding of safety and our trust in the safety arguments for a system. Gribaudo et al in [4] propose a methodology to link functional analysis together with stochastic analysis. The proposed methodology relies on the representation of systems by Fluid Petri Nets (FPNs). They show how FPNs can be used as input for functional analysis based on model checking as well as stochastic analysis. A case study drawn from a control system shows the applicability of the approach. A stochastic analysis first uses the FPN system model in order to evaluate the system performance. Functional analyses of the system rely on model checking

[1] http://www.ewics.org/.
[2] http://www.safecomp.org/.
[3] Recent SAFECOMP proceedings have been published by Springer-Verlag in the series LNCS. They are accessible online via the SAFECOMP web site or directly via the Springer LNCS web site.

doi:10.1016/S0951-8320(03)00088-7

techniques. The experiment assesses three different model checkers: HyTech, NuSMV, and PRISM. In order to model check the system using HyTech, the FPN system model needs to be translated in a linear hybrid automata (LHA). This is because HyTech can only process system models that define continuous dynamics by constant limits on time derivatives. If the model is more complex, it needs to be split into simpler submodels having constant limits on time derivatives. In the worst case the analysis by Hytech is exponential in space and time with respect to the model size. Hence HyTech is suitable for small LHA. In contrast, NuSMV can process discrete models directly drawn from FPNs. The main limitation is that NuSMV can only process finite state systems. In the worst case the analysis by NuSMV is exponential in space and time with respect to the model size. Hence NuSMV is suitable for moderate-size finite state models. Finally, PRISM uses discrete time markov chains (DTMCs). It performs probabilistic model checking of system properties. In the worst case the analysis by PRISM is exponential in space and time with respect to the model size. It is therefore suitable to perform exact probabilistic analyses of moderate-size DTMCs. In summary, the results in [4] show how to combine different methodologies in order to perform both functional and performance analyses. This experience points out the limits of current methodologies and draws directions for future research.

Ortmeier et al in [5] show how to combine model checking and Fault Tree Analysis (FTA) in order to perform safety analysis. They applied the combined safety analysis on a control system drawn from the Elbtunnel project. Model checking assesses safety properties of the system. The formal specification of the system consists of timed automata with a finite set of states. The automata described in [5] correspond to the formal specifications in the model checker RAVEN. In addition, FTA analyses potential system failures for given hazards. The combination of model checking and FTA identifies safety flaws in the system design. This analysis led to proposals for changes in the design in order to improve the over all system safety. The safety analysis then consists of two different methodologies, i.e. model checking and FTA, applied in isolation. The results of these two analyses are then integrated together. This suggests that it might be interesting to investigate process diversity for safety analysis in future research. In other words, diversity may enhance our ability to perform safety analysis by model checking and FTA. The results in [5] show how early phases, in particular, requirements specification, in the system life cycle can benefit by combinations of different methodologies. The discovery of requirements flaws early in the development process is useful in order to reduce the risk associated with faulty requirements and the cost of fixing them.

Bobbio et al in [6] show how the combination of heterogeneous stochastic model techniques increases modeling expressiveness, which enhances our ability to assess system safety. A Fault Tree (FT) models a system drawn from a digital control system. The assumptions underlying the FT model are that each component has a binary (i.e. up and down) behavior and that failures are statistically independent. FT models any relation between failures and causes by logical (i.e. AND, OR and in some cases XOR) operators. The FT model then allows us to perform a statistical safety analysis of the system and to compute safety related measures (e.g. Safe Mission Time, Mean Time To Failure, etc.). Translating the FT model in a Bayesian Network (BN) may further enrich the safety analysis. The translation from FT to BN can be performed automatically. The BN model extends the FT model by capturing probabilistic logical operators, multi-state variables and sequentially dependent failures. In other words, the BN model takes into account non-binary events and localized dependencies between components. Furthermore, this allows us to perform a posterior analysis of the criticality of each system component. Although the translation to a BN model enhances our ability to model complex system failure modes, it fails to capture cyclic system behaviors. For instance, it fails to model actions that bring the system in a previous state. Recover actions like maintenance or fault tolerance typically take the system into an early state.

Translating the FT model to Stochastic Petri Nets (SPNs) allows us to model these cases as well. In order to limit the state explosion problem it is possible to adopt a similar approach that translates FT to a special class of SPNs, named, Stochastic Well-formed Nets (SWNs). This model enrichment allows repair actions to be taken into account and allows availability analysis of the case study. Finally, Bobbio et al in [6] show how model expressiveness is important in order to assess system safety. Furthermore, their experience points out how safety analysis may require subsequent refinements through different models that capture different system properties. Hence we can conclude that the choice of the model must be carefully evaluated according to the stringency of the system requirements. Future research should investigate how to support the choice of the modeling techniques. Guidelines on how to select specific models and industrial usages may stimulate further increased industrial adoption of formal methods.

The verification and validation of system software represent other traditional fields in which the application of formal methods is beneficial. Although research and practice have extensively investigated the use of formal methods to verify and validate system software. Formal methods are not fully integrated into industrial verification and validation. Two main issues limit our ability effectively to use formal methods. The first is our limited ability to reuse formal arguments. The second is the lack of incremental approaches for the verification and validation of system software using formal methods. Sharma et al in [7] describe an Assertion Checking Environment (ACE) for the compositional verification of MISRA C programs.

Assertions are derived from the specification of the program units. ACE supports verification of static assertions, which can be performed without the execution of the program units. ACE is interfaced to the STeP theorem proved in order to verify the assertions in the program. The results show that compositional verification may limit the cost of verifying software programs by verifying small program units with limited complexity.

Decomposing system software to small and manageable entities is essential in order effectively to support system design process phases, from design to certification. Nowadays an increasing number of systems consists of Commercial Off-The-Shelf (COTS) software. This aims to increase software reusability across industrial contexts as well as to reduce software cost. In the long term it is believed that COTS software may increase the overall system dependability. An increasing business demand for cheaper and more dependable software has supported the popularity of COTS software. Thus system dependability relies on (outsourcing) COTS software. New issues arise in the certification of COTS software that relies on the trustworthiness of third parties. Bishop et al describe in [8] a Software Criticality Analysis (SCA) developed in order to support the use of COTS software in safety-related systems. The SCA methodology aims to assess the extent to which each component contributes to safety of the whole system. The SCA methodology furthermore points out segregation between software components with different safety importance. The methodology consists of a combination of Hazops based on design documents and software inspections. The results point out that the assessment based only on architecture and design document would have been misleading.

Bate and Kelly introduce in [9] a modular approach for the construction of safety cases. The construction of safety cases relies on architectural considerations in order to improve the flexibility of function allocation as well as maintainability and to reduce development costs. The proposed approach relies on two major properties, namely, segregation and location independence. These allow integrating architectural considerations in the constructions of safety cases.

Papadopoulous [10] proposes a methodology for a safety monitor, which relies on information provided by Statecharts and FTs. These methodologies are classically used during design and certification. The proposed methodology exploits these models and integrates them together with usage information. The resulting model-based approach may support real-time detection, diagnosis and control of hazardous failures. The approach is assessed by a case study of an online safety monitor.

The papers selected for this special issue highlight challenges in the development of dependable systems. Future research should further investigate our ability in integrating different approaches. The selected papers point out that the integration of different methodologies is still a trick that requires considerable effort and skilled expertise. Future research should also assess our ability to transfer specific technology to industrial contexts by integrating specific methodologies into work practice. This requires further integrating methodologies (formal as well as less-formal) throughout the system life cycle. The role of certification activities and standards still remain central in the trustworthiness of dependable systems. The systems arising in the modern electronic mediated society may require different model of trust. Unfortunately, multi-disciplinary work on human factors in dependable systems is still scarce and patchy. Additional effort must be spent in cross-fertilizations between different research communities. Classical engineering methodologies fail to capture human factors. Socio-technical systems give rise to new failure modes that have origins in lack of understanding of human factors as well as in the emergent and evolving nature of these systems. On the other hand methodologies grounded in human related theories (e.g. Cognitive Science, Social Science, Activity Theory, Distributed Cognition, Situation Awareness, etc.) need to perform a step forward towards engineering methodologies. Recent research clearly draws future directions. Although, as we mentioned earlier, we are still far away from the Renaissance of the modern electronically mediated society, the next industrial revolution may be just around the corner.

A three-step process has selected the papers forming this special issue. The SAFECOMP 2002 International Program Committee has selected a subset of potential best papers. The authors were then invited to submit an extended and revised version of their SAFECOMP 2002 papers. A second review process was then arranged in order to review the extended and revised versions of the papers. This allowed the authors to provide further technical details without the length limit that constrains the proceedings papers. The final selection for this special issue was then made according to the initial reviews for the SAFECOMP 2002 papers and the additional reviews on the extended and revised versions of the papers. The authors of the accepted papers were then required to take into account the additional comments in order to prepare the final version for this special issue. This selection process would have been impossible without the effort and support by the SAFECOMP 2002 International Program Committee. We have been furthermore able to speed-up the entire review process tanks to the prompt collaboration of the authors. We would like to thank the SAFECOMP 2002 International Program Committee for the tremendous work, the authors for their excellent papers and cooperation. Special thanks go to the editor-in-chief of this journal, Prof. G.E. Apostolakis, for his support throughout the entire process.

References

[1] Anderson S, Bologna S, Felici M, editors. Computer safety, reliability and security. Proceedings of the 21st International Conference,

SAFECOMP 2002, Catania, Italy, September 10–13, LNCS 2434, Springer; 2002.
[2] Kanoun K, Pasquini A. Guest editorial: safety, reliability and security of industrial computer systems. Reliab Engng Syst Safe 2001;71(3): 227–8.
[3] van der Meulen M, Koornneef F. Guest editorial: safety, reliability and security of industrial computer systems. Saf Sci 2002;40(9): 715–7.
[4] Gribaudo M, Horváth A, Bobbio A, Tronci E, Ciancamerla E, Minichino M. Fluid petri nets and hybrid model-checking: a comparative case study. Reliab Engng Syst Saf 2003; 81(3): 239–257.
[5] Ortmeier F, Schellhorn G, Thumus A, Reif W, Hering B, Trappschuh H. Safety analysis of the height control system for the Elbtunnel. Reliab Engng Syst Saf 2003; 81(3): 259–268.
[6] Bobbio A, Ciancamerla E, Franceschinis G, Gaeta R, Minichino M, Portinale L. Sequential application of heterogeneous models for the safety analysis of a control system: a case study. Reliab Engng Syst Saf 2003; 81(3): 269–280.
[7] Sharma B, Dhodapkar SD, Ramesh S. Assertion checking environment (ACE) for formal verification of C programs. Reliab Engng Syst Saf 2003; 81(3): 281–290.
[8] Bishiop P, Bloomfield R, Clement T, Guerra S. Software criticality analysis of COTS/SOUP. Reliab Engng Syst Saf 2003; 81(3): 291–301.
[9] Bate I, Kelly T. Architectural considerations in the certification of modular systems. Reliab Engng Syst Saf 2003; 81(3): 303–324.
[10] Papadopoulous Y. Model-based system monitoring and diagnosis of failures using state charts and fault trees. Reliab Engng Syst Saf 2003; 81(3): 325–341.

Stuart Anderson
Massimo Felici*
LFCS, School of Informatics,
The University of Edinburgh, Mayfield Road,
Edinburgh EH9 3JZ, Scotland, UK
E-mail address: massimo.felici@ed.ac.uk

* Corresponding author. Tel.: +44-131-6505899; fax: +44-131-6677209.