Formal Engineering of Resilient Systems: Achievements and Challenges

Elena Troubitsyna
Åbo Akademi University
Turku, Finland
Email: Elena.Troubitsyna@abo.fi

Alexander Romanovsky
Newcastle University,
Newcastle, UK
Email: Alexander.Romanovsky@ncl.ac.uk

***Abstract*—In this paper we overview the work on formal engineering of resilient systems carried out in the FP7 EU Deploy project. We discuss our experience and open issues.**

***Keywords*-resilience; formal modelling; Event-B**

I. Introduction

Resilience - the ability of a system to deliver services that can be justifiably trusted despite changes [1] - is an evolution of the dependability concept. It encompasses the system aptitude to autonomously adapt to evolving requirements, operating environment changes and/or component failures. To design resilient systems, we need scalable modelling techniques that can cope with complexity, explicitly address various aspects of resilience throughout the entire development cycle and ensure that the system adapts to changes safely. In this paper we overview some of the work on modelling of resilient systems that we have carried out in the FP7 EU Deploy project [2].

II. Refinement approach to development of resilient systems

It is widely recognised that system complexity can be managed via abstract modelling, decomposition and iterative development. Event-B [3] is a formal top-down development approach to correct-by-construction system development. Development in Event-B starts from defining a high-level specification that represents the system behavior and properties in a highly abstract way. The main development technique – refinement – allows us to ensure that a concrete specification preserves the globally observable behaviour and properties of the abstract specification. Verification of each refinement step is done by proofs. The Rodin platform [4] automates modelling and verification in Event-B.

Event-B has been actively used within the EU Deploy project [2] to model resilient systems from various application domains. As a result, we have created a number of formal approaches to explicitly reason about resilience in the refinement process.

III. Goal-oriented refinement of reconfigurable systems

In [5], [6], we have investigated the problem of ensuring interoperability of agents in mobile agent systems. The work has resulted in defining the modelling patterns to represent agent roles in dynamic scopes and the logical conditions to ensure agent compatibility.

In [7], we have continued our study of multi-agent systems and have proposed a goal-oriented approach to development of multi-agent systems. Essentially, our approach allows the developers to define system goals at different levels of abstraction and guarantee goal reachability despite agent failures. We have derived refinement patterns modelling the mechanisms for dynamic system reconfiguration by reallocating goals from failed agents to healthy ones and, per se, guarantee resilience.

While refining a reconfigurable system, we had to assume that sufficient amount of agents would remain operational to achieve the desired goals. In [8], we have demonstrated how to integrate probabilistic analysis to quantitatively assess the likelihood of goal reachability despite failures. The rigorous refinement process has allowed us to establish the precise relationships between component failures and goal reachability. We have assessed the derived reconfigurable system architecture to quantitatively verify that it achieves the desired reliability and performance objectives. This was accomplished by relying on the probabilistic extension of Event-B to verify reliability and performance properties using PRISM model checker [9].

IV. Formal development of fault tolerant mode-rich systems

A widely used mechanism for adapting to changing operating conditions is based on the notion of modes. In our work [10], [11], [12], we have proposed an approach to formal development of fault tolerant mode-rich systems. Such systems achieve fault tolerance by rollbacking to specific degraded modes. The proposed formal development process allows the designers to develop a system in a layered fashion. Essentially, it consists of a number of steps gradually unfolding system architectural layers by refinement. Moreover, we prove the consistency between mode transitions on adjacent architectural layers, which significantly improves scalability of verification.

In our approach to modelling mode-rich systems, we have focused on verification of consistency of a predefined mode logic. In [13], we have proposed to conduct Failure Modes and Effects Analysis (FMEA) of each operational

mode to identify mode transitions required to implement fault tolerance. Fault tolerance is achieved by two different means – transitions to a more degraded mode and dynamic reconfiguration using redundant components. Furthermore, we have investigated a complex interplay between the states of components during reconfiguration and the system modes.

V. Integrating safety analysis into formal development

In [14], we have demonstrated how to combine formal modelling and refinement with Failure Modes and Effects Analysis (FMEA). We have defined a set of patterns formalising the requirements derived from FMEA as well as automated their integration into the formal specification. The proposed approach facilitates formal development and improves traceability of safety requirements.

The systems, whose components are susceptible to various kinds of faults, never are "absolutely" safe, i.e., certain combinations of failures may lead to an occurrence of a hazard – a potentially dangerous situation breaching safety requirements. To demonstrate that the probability of a hazard occurrence is acceptably low, in [15] we have presented a formal approach to integrating quantitative safety analysis into formal system development by refinement in Event-B. Essentially, our approach can be seen as a process of extracting a fault tree – a logical representation of a hazardous situation in terms of the primitives used at different abstraction layers. Eventually, we arrive at the representation of a hazard in terms of the failures of basic system components, which allows us to calculate probability of a hazard occurrence.

VI. Discussion

Our work on formal engineering of resilient systems in the EU Deploy project has resulted in two types of approaches:

- the approaches that focus on creating modelling patterns and guidelines for representing and verifying certain resilience-related behavior
- the approaches that integrate (external) techniques for safety and reliability analysis into the formal development process of Event-B.

A tight cooperation with the Deploy industrial partners has allowed us to gain rich experience in modelling resilient systems from the transportation, aerospace and business information system domains. The development of industrial-scale systems has emphasized the need for scalability in formal modelling and automatic tool support. It has fostered the research on modularisation and decomposition techniques for Event-B as well as development of various plug-ins, e.g., see [16]. Moreover, it has led to understanding importance of heterogenous modelling techniques to address a variety of resilience aspects.

In general, we believe that Event-B offers a powerful formal technique for engineering resilient systems. To leverage scalability and industrial relevance of the method, we will continue to enlarge the set of modelling patterns for representing various resilience aspects, strengthening automatic tool support and enriching its capabilities via dedicated plug-ins to the Rodin platform.

References

[1] J.-C. Laprie, "Resilience for the scalability of dependability," in *NCA*, 2005, pp. 5–6.

[2] Industrial Deployment of System Engineering Methods Providing High Dependability and Productivity (DEPLOY), "IST FP7 IP Project, online at http://www.deploy-project.eu/."

[3] J.-R. Abrial, *Modeling in Event-B*. Cambridge University Press, 2010.

[4] Rodin, "Event-B Platform, online at http://www.event-b.org/."

[5] L. Laibinis, E. Troubitsyna, A. Iliasov, and A. Romanovsky, "Formal Approach to Ensuring Interoperability of Mobile Agents," in *Handbook of Research on Mobile Software Engineering: Design Implementation and Emergent Applications*, P. Alencar and D. Cowan, Eds. IGI Global, 2011.

[6] I. Pereverzeva, E. Troubitsyna, and L. Laibinis, "Formal Development of Critical Multi-Agent Systems: A Refinement Approach," in *EDCC-9. European Dependable Computing Conference*, M. Correia, Ed. IEEE CPS, 2012, To appear.

[7] ——, "Formal Goal-Oriented Development of Resilient MAS in Event-B," in *Ada-Europe 2012. 17th International Conference on Reliable Software Technologies*, M. Brorsson and L. M. Pinho, Eds. Springer-Verlag, 2012, To appear.

[8] A. Tarasyuk, I. Pereverzeva, E. Troubitsyna, T. Latvala, and L. Nummila, "Formal Development and Assessment of Reconfigurable On-Board Satellite System," *TUCS Technical Reports 1038*, 2012.

[9] PRISM, "Probabilistic symbolic model checker, online at http://www.prismmodelchecker.org/."

[10] A. Iliasov, E. Troubitsyna, L. Laibinis, A. Romanovsky, K. Varpaaniemi, D. Ilic, and T. Latvala, "Developing mode-rich satellite software by refinement in event b," in *FMICS*, 2010, pp. 50–66.

[11] A. Iliasov, E. Troubitsyna, L. Laibinis, A. Romanovsky, K. Varpaaniemi, P. Väisänen, D. Ilic, and T. Latvala, "Verifying mode consistency for on-board satellite software," in *SAFECOMP*, 2010, pp. 126–141.

[12] A. Iliasov, E. Troubitsyna, L. Laibinis, A. Romanovsky, K. Varpaaniemi, D. Ilic, and T. Latvala, "Developing mode-rich satellite software by refinement in event-b," *Science of Computer Programming*, 2012. To appear.

[13] Y. Prokhorova, L. Laibinis, E. Troubitsyna, K. Varpaaniemi, and T. Latvala, "Derivation and formal verification of a mode logic for layered control systems," in *APSEC*, 2011, pp. 49–56.

[14] I. Lopatkin, A. Iliasov, A. Romanovsky, Y. Prokhorova, and E. Troubitsyna, "Patterns for representing fmea in formal specification of control systems," in *HASE*, 2011, pp. 146–151.

[15] A. Tarasyuk, E. Troubitsyna, and L. Laibinis, "Quantitative Verification of Safety in Event-B," in *SERENE 2011, Software Engineering for Resilient Systems*. Springer-Verlag, 2011, pp. 24–39.

[16] I. Lopatkin, A. Iliasov, and A. Romanovsky, "Rigorous development of dependable systems using fault tolerance views," in *ISSRE*, 2011, pp. 180–189.