ISSN 2658-5014 (Print)
2019. № 1. С. 41–46
СИИТ
СИСТЕМНАЯ ИНЖЕНЕРИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
http://siit.ugatu.su

УДК 004.65

ANALYTICAL TOOLSET FOR MODEL-BASED STOCHASTIC ERROR PROPAGATION ANALYSIS: EXTENSION AND OPTIMIZATION TOWARDS INDUSTRIAL REQUIREMENTS

T.I. FABARISOV[1], N. I. YUSUPOVA[2]
K. DING[3], A. MOROZOV[4], K. JANSCHEK[5]

[1] flatagir@gmail.com, [2] yussupova@ugatu.ac.ru, [3] kai.ding@tu-dresden.de, [4]andrey.morozov@tu-dresden.de, [5]klaus.janschek@tu-dresden.de

[1,2] ФГБОУ ВО «Уфимский государственный авиационный технический университет» (УГАТУ)

[3–5] Technische Universität Dresden (Germany)

Поступила в редакцию 20 декабря 2018 г.

Abstract. Model-Based System Engineering (MBSE) is a popular mathematical and visual approach to the design of complex control, signal processing, and communication systems. It is used in safety critical industrial domains including aerospace, automotive, transportation, medical and robotics applications. Our group develops methods and tools for model-based system reliability and safety analysis with the main focuses on stochastic modelling of error propagation processes. This article is devoted to the optimisation and extensions to our analytical toolset. We have investigated the key modeling paradigms, requirements and industrial needs and have formulated the list of particular extensions.

Keywords: Error propagation model; reliability; safety; dependability; model-based systems; model-based analysis; control flow; data flow; optimization.

INTRODUCTION

Model-Based System Engineering (MBSE) [1] is a methodology that focuses on problems associated with designing of complex control, signal processing, and communication systems. It is used in motion control, industrial equipment, aerospace, and automotive applications.

The MBSE methodology aims at the increasing of productivity [21] and, as the result, speeding up the development process. In order to overcome the gap between the system model's properties and simulation software, we need a methodology to cover various technical aspects related to use of the model simulation software. Therefore, the term Model-Based System Engineering has started to use alongside with the term Modeling and Simulation-based Systems Engineering (M&SBSE).

The MBSE approach is based on the creating, reusing and exploiting of the already available domain models. This allows a designer to reuse existing well tested functional blocks and interfaces, which generally leads to avoiding the common errors that the designer otherwise could have made. This also helps to save time and human-hours resources which leads to the increasing cost efficiency and allows to focus on the general system design rather than the development of the particular system parts.

Furthermore, MBSE methodology is the modern trend for safety-critical industrial domains, where a failure or malfunction may result in environmental harm, severe equipment damage or even serious injuries of the personal.

Model-based analysis is a methodology for the analysis that is based on models that represent the desired system which is currently under development. Base-line system models can be used not only for further system development, but also for system analysis on earlier design phases. This helps to find and fix the errors, shortcomings and drawbacks on the earlier phases without a need of developing the whole system and before the implementation. Model-based system reliability and safety anal-

Работа поддержана грантом Erasmus+

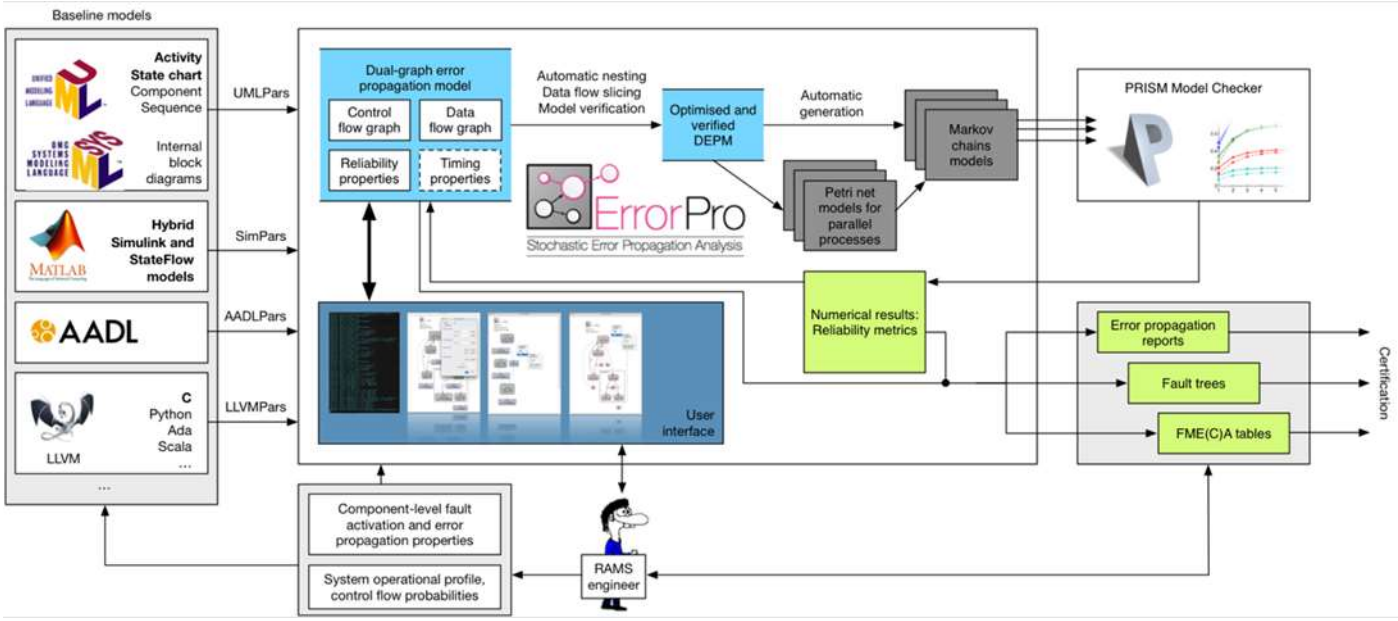

Fig. 1. Analytical workflow of the EPA

ysis are important due to the fact that it affects both the utility and the life-cycle costs of a product or complex systems, where the failure or malfunction may result in serious injury (fig. 1).

ERROR PROPAGATION ANALYSIS

Model-Based System Engineering (MBSE) [1] is a methodology that focuses on problems associated with designing of complex control, signal processing, and communication systems. It is used in motion control, industrial equipment, aerospace, and automotive applications.

DUAL-GRAPH ERROR PROPAGATION MODEL

The Dual-graph Error Propagation Model (DEPM) is a mathematical abstraction [2] of the main future system's properties, which are vital for the determination of the error propagation processes. It is a useful analytical instrument for the evaluation of the influence of particular faults and errors to the overall system behavior.

The following description of fig. 2 is the set-based mathematical notation of the target dual-graph error propagation model:

$$DEPM := \langle E, D, ACF, ADF, C \rangle$$

E is a non-empty set of elements;

D is a set of data storages;

ACF is a set of directed control flow arcs, extended with control flow decision probabilities;

ADF is a set of directed data flow arcs;

C is a set of conditions of the elements.

A simple DEPM is shown in fig. 2. Elements A, B, and C represent executable parts of the system. Each element has a zero or more data inputs and outputs. Data storages M1, M2, M3, M4, CM1 represent variables, which can be read or written by the elements. During execution of an element, e.g. element A that is highlighted in red, errors can occur and propagate to its data outputs. The incoming errors can propagate from the inputs to the outputs depending on the internal properties of the element, defined with the probabilistic conditions.

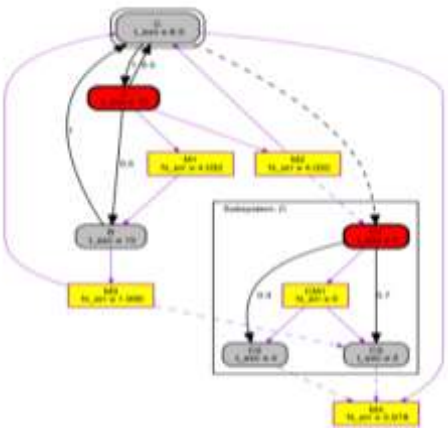

Fig. 2. An example of a simple dual-graph error propagation model

The control flow arcs (black lines) connect the elements. Each control flow arc is weighted with a transitions probability: After the execution of A, B will be executed with probability 0.5 and C with probability 0.5.

The data flow arcs (purple lines) describe data transfer between the elements. A data flow arc connects an element with a data storage or vice versa. The data flow arcs are considered to be the paths of data error propagation.

ANALYTICAL TOOLSET

The ErrorPro™ [3] is our analytical software that supports the system analysis with the DEPM. Figure 1 demonstrates the intended analytical workflow. On the input we have the baseline models that describe the system. We developed several parsers that can transform the baseline models into the dual-graph error propagation models which consist of Control flow graph, Data flow graph, Reliability properties, and Timing properties. This DEPM is also shown in the graphical user interface. The ErrorPro™ supports automatic nesting, data flow slicing and model verification. Then from the optimized and verified DEPM the ErrorPro™ automatically generates Markov chains and Petri net models for parallel processes which can be checked by the third-party software PRISM Model Checker [4]. The computed numerical reliability metrics are both shown to the user in the GUI and used for the generation of the system reliability reports, which can be used for the forthcoming certification.

ErrorPro™ plays a role of an intermediate analytical model between the main baseline system model and the mathematical model that can be formally analyzed. This allows the system optimizations on a higher level of abstraction that helps to avoid the typical issue of the state based models - state space explosion: As the number of states of the Markov chain model grows exponentially with the growth of the size level of detail of the baseline model. The ErroPro™ uses several built-in optimization algorithms for data flow slicing and automatic model nesting. Also, the ErrorPro™ works as an integrator that allows the analysis of the systems that are based on various baseline models e.g. when UML or AADL are used for the high-level design and the particular functional blocks are implemented with Simulink/Stateflow models.

ErrorPro™ is a software tool for stochastic error propagation analysis, based on the described DEPM. This tool automatically creates discrete time Markov chain models using the DEPM representation and computes required numerical reliability properties. A DEPM model can be stored in an XML file. A fragment of an XML file of the DEPM from fig. 3 is shown in List.1.

```
<model n_steps="100" name="abc" version="3.0">
<element execution_time="10.0" initial="true"
name="A"/>
<control_flow from="A" prob="0.5" to="B"/>
<control_flow from="A" prob="0.5" to="C"/>
<data_flow from="A" to="M1"/>
<data_flow from="A" to="M2"/>
<conditions element_name="A" sta-
tus="acknowledged">
<if statement="True">
<then prob="0.9" update="data_state['M1'] = 'ok';
data_state['M2'] = 'ok'"/>
<then prob="0.1" update="data_state['M1'] = 'er-
ror'; data_state['M2'] = 'error'"/>
</if>
</conditions>
</model>
```

List. 1. An example of a DEPM XML file

ADVANTAGES OF THE OPENERRORPRO

A number of other tools and methods exist and can be used for the solution of similar tasks. Most of them are falling into the R&D domain "Model Checking". Low-level stochastic model checkers such as PRISM or MRMC [5] can model the error propagation processes, but the direct generation of the Markov models from the baseline models are not trivial and may easily result in the explosion of the state space. The high-level model checkers oriented exactly to the reliability and safety analysis such as COMPASS [6] or AltaRica [7]. On the one hand, our approach is more specific than COMPASS and AltaRica since it is developed only for the stochastic error propagation analysis.

The discussed model checkers have very broad analytical focus and, therefore, requires

considerable manual manipulations with the baseline models for the modeling of the error propagation as well as the deep knowledge of probabilistic model checking methods, Markov models, and temporal logic. On the other hand, our approach is not limited to the AADL (SLIM) or the AltaRica language and can be applied even to mixed model-based systems e.g a top-level AADL or SysML model and Simulink implementations of particular components.

Last but not least, our parsers allow the automatic generation of the DEPMs from the various baseline models including UML/SysML [8, 20], AADL [17], Simulink/Stateflow [18] and source code via the LLVM [19] technology.

CASE STUDIES

Our methods have been applied for several case studies during the last years. Different types of mechatronic systems have been analyzed, including the moving and flying robots, and specific model-based software systems from the space and automotive industrial domains. Stochastic analysis of critical error propagation paths and probabilistic distributions of various faulty and fault-free system execution scenarios is presented in [9, 10].

Estimation of the likelihood of error propagation to the critical parts (outputs) of systems modeled with Simulink and Stateflow is presented in [11]. Application of the error propagation analysis for the automatic prioritization of test cases for effective testing of Simulink/Stateflow models after minor updates is shown in [12]. Estimation of system reliability properties under the occurrence of timing errors caused by faulty synchronization of parallel processes is discussed in [13, 14]. An approach for the balancing of performance vs. reliability and for system protection with software-implemented hardware fault detectors is introduced in [15].

REQUIRED EXTENSIONS

We have investigated the key modeling paradigms, requirements and industrial needs in order to formulate the list of necessary extensions both for our analytical approach and the tool set.

Parallel model: Nowadays safety critical systems consist of heteroscedasticity distributed components. This results in parallel processes and complex error propagations patterns. The current version of the ErrorPro supports only sequential systems. The transition from the sequential to the parallel models requires considerable effort and might bring additional computational complexity. However, this is required by the error propagation analysis of distributed and heterogeneous systems especially taking into account the current trend to the Internet of Things and cyber-physical systems.

Timing: Timing plays an important role in system design. A timing error can appear in the system for example when the parallel processes are not perfectly synchronized. The data can be transmitted either too early or too late. This can lead to dangerous consequences. The nominal way of the safety critical system design is the application of hard real time hardware and software. However, nowadays the complexity and heterogeneity of the systems are dramatically increasing and it is required to analyze and verify the safety of the systems that contain non-real-time components. Therefore, it is necessary to extend the DEPM elements with the timing information and take this into account during the analysis.

Nesting: The solutions for the state explosion problem are model slicing and automatic model nesting. Partially these functionally are already implemented in the ErrorPro. This allows the generation of several simple and computable Markov models instead of one gigantic and non-computable Markov model. An improved algorithm for automatic nesting has been presenting in [16]. This algorithm will be further improved and embedded in the new version.

Monte Carlo simulation: The current version is based on the underlying Markov chain models. However due to the discussed state explosion problem, even after the application of all possible optimization methods, sometimes the only possible solution is the Monte Carlo simulation. We plan to extend the tool with the simulation capabilities. This will also help to verify the analytical methods.

Python 3.0: Technically ErrorPro™ is implemented in Python 2.7. But due to the limita-

tions, it will be not supported in the next years. It should be translated to the Python 3.0 in order to cease the bugs and limitations inherited from the old version of the Python.

Multiple error types: The current version works with the simple binary representation of errors. Basically, the result is there an error or there is no error. We are planning to extend the ErrorPro™ towards the support of multiple types of errors. This will significantly increase the modulating capability of the tool.

Report generation: The industrial standards require an application of the specific method for the reliability calculations such as fault tree analytics (FTA) and failure mode and effect analytics (FMEA). We will extend the tool with the methods for auto generation FTA and FMEA based on the given DEPMs.

All the extensions of the ErrorPro™ will, of course, lead to the changes of the parsers that also will be implemented.

CONCLUSION

A software tool ErrorPro™ for stochastic error propagation analysis, based on the described DEPM has been discussed in this article. This tool automatically creates discrete time Markov chain models using the DEPM representation and calculates required numerical reliability properties.

ErrorPro™ plays a role of an intermediate analytical model between the main baseline system model and the formal mathematical model that can be formally analyzed. This allows the system optimizations on a higher level of abstraction that helps to avoid the typical problem of the state based models - state space explosion.

The key modeling paradigms, requirements and industrial needs have been formulated into the list of particular extensions of the functionality that has been elaborated in this article.

REFERENCES

1. **Gianni, Daniele; D'Ambrogio, Andrea; Tolk, Andreas, eds.** Modeling and Simulation-Based Systems Engineering Handbook (1 ed.). USA: CRC Press, 2014.
2. **A. Morozov and K. Janschek.** Dual graph error propagation model for mechatronic system analysis. In 18th IFAC World Congress, Milano, Italy, pp. 9893-9898, 2011.
3. **A. Morozov, R. Tuk, and K. Janschek.** ErrorPro: Software Tool for Stochastic Error Propagation Analysis. In 1st International Workshop on Resiliency in Embedded Electronic Systems, Amsterdam, The Netherlands, pp. 59-60, 2015.
4. **Marta Kwiatkowska, Gethin Norman and David Parker.** PRISM 4.0: Verification of Probabilistic Real-time Systems. In Proc. 23rd International Conference on Computer Aided Verification (CAV'11), volume 6806 of LNCS, pp. 585-591, Springer, 2011.
5. **Katoen J P, Zapreev I S, Hahn E M, et al.** The ins and outs of the probabilistic model checker MRMC[J]. Performance evaluation, 2011, 68(2): 90-104.
6. **Bozzano M, Cimatti A, Katoen J P, et al.** The COMPASS Approach: Correctness, Modelling and Performability of Aerospace Systems[C]//SAFECOMP. 2009, 5775: 173-186.
7. **Arnold A, Point G, Griffault A, et al.** The AltaRica formalism for describing concurrent systems[J]. Fundamenta Informaticae, 1999, 40(2, 3): 109-124.
8. **OMG.** OMG Unified Modeling Language (OMG UML), 2015.
9. **A. Morozov and K. Janschek.** Case study results for probabilistic error propagation analysis of a mechatronic system, In: Tagungsband VDI Fachtagung Mechatronik 2013, Aachen, 06.03.-08.03.2013, pp. 229-234.
10. **A. Morozov and K. Janschek.** Probabilistic error propagation model for mechatronic systems. Mechatronics, 24(8):1189 – 1202, 2014.
11. **A. Morozov, K. Janschek, T. Krüger, A. Schiele.** Stochastic Error Propagation Analysis of Model-driven Space Robotic Software Implemented in Simulink. In Proceedings of the Third Workshop on Model-driven Robot Software Engineering, Leipzig, Germany, 2016.
12. **Morozov, A., Ding, K., Chen T. and Janschek, K.** Test Suite Prioritization for Efficient Regression Testing of Model-based Automotive Software. Accepted paper. Proceedings of the annual conference on Software Analysis, Testing and Evolution (SATE), Harbin, China, 2017.
13. **K. Ding, T. Mutzke, A. Morozov, and K. Janschek.** Automatic Transformation of UML System Models for Model-based Error Propagation Analysis of Mechatronic Systems. In Proceedings of 7th IFAC Symposium on Mechatronic Systems, Loughborough University, UK, 2016.
14. **T. Mutzke, K. Ding, A. Morozov, K. Janschek, J. Braun.** Model-based Analysis of Timing Errors for Reliable Design of Mechatronic Medical Devices, In Proceedings of 3rd International Conference on Control and Fault-Tolerant Systems, Barcelona, Catalonia, 2016.
15. **A. Morozov, K. Janschek.** Flight Control Software Failure Mitigation: Design Optimization for Software-implemented Fault Detectors. In Proceedings of 20th IFAC Symposium on Automatic Control in AerospaceACA 2016 — Sherbrooke, Quebec, Canada, 21-25 August 2016.
16. **Zhao F, Morozov A, Yusupova N I, et al.** Nesting algorithm for dual-graph error propagation models[C]//CSIT'2016. 2016:p.p.106-110.
17. **B. Lewis, P. Feller.** Impact of Architectural Model-Based Engineering with AADL, Carnegie Mellon University, 2007.
18. **Zou L., Zhan N., Wang S., Fränzle M.** Formal Verification of Simulink/Stateflow Diagrams. In: Finkbeiner B., Pu G., Zhang L. (eds) Automated Technology for Verification and Analysis. Lecture Notes in Computer Science, vol 9364. Springer, Cham, 2015.
19. **Chris Lattner Vikram Adve.** LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. University of Illinois at Urbana-Champaign.

20. **Delligatti, Lenny. SysML Distilled**. A Brief Guide to the Systems Modeling Language. — Addison-Wesley Professional, 2013.

21. **Morozov, Andrey & Janschek, Klaus & Krüger, Thomas & Schiele, André.** (2016). Stochastic Error Propagation Analysis of Model-driven Space Robotic Software Implemented in Simulink. MORSE '16. Proceedings of the 3rd Workshop on Model-Driven Robot Software Engineering: 24-31.

METADATA

Title: analytical toolset for model-based stochastic error propagation analysis: extension and optimization towards industrial requirements.

Authors: T.I. Fabarisov[1], N. I. Yusupova[2], K. Ding[3], A. Morozov[4], K. Janschek[5]

Affiliation:

[1,2] Ufa State Aviation Technical University (UGATU), Russia.
[3–5]Technische Universität Dresden, Germany.

Email:

[1]flatagir@gmail.com,[2]yussupova@ugatu.ac.ru,
[3]kai.ding@tu-dresden.de,[4]andrey.morozov@tu-dresden.de,
[5]klaus.janschek@tu-dresden.de.

Language: English.

Source: SIIT, no. 1, pp. 41-46, 2019. ISSN 2658-5014 (Print).

Abstract: Model-Based System Engineering (MBSE) is a popular mathematical and visual approach to the design of complex control, signal processing, and communication systems. It is used in safety critical industrial do-mains including aerospace, automotive, transportation, medical and robotics applications. Our group develops methods and tools for model-based system reliability and safety analysis with the main focuses on stochastic modelling of error propagation processes. This article is devoted to the optimization and extensions to our analytical toolset. We have investigated the key modeling paradigms, requirements and industrial needs and have formulated the list of particular extensions.

Key words: Error propagation model, reliability, safety, dependability, model-based systems, model-based analysis, control flow, data flow, optimization.

About authors:

FABARISOV, Tagir Ildarovich, PhD Student, Dept. of Computational Mathematics and Robotics. Master in computer science (UGATU, 2018).

YUSUPOVA, Nafisa Islamovna, prof., head of Dept. of Computational Mathematics and Robotics. Dipl. Radiophysicist (Voronezh State Univ. 1975). Dr. Techn. (USATU, 1998).

DING, Kai, Ph.D. student at the Institute of Automation at the Faculty of Electrical and Computer Engineering, Technische Universität Dresden, Germany. He received his Dipl.-Ing. degree at Technische Universität Dresden in 2016. His research interests include fault-tolerant systems, dependability design patterns, software reliability and model-based systems analysis.

MOROZOV, Andrey, postdoc researcher at the Institute of Automation at the Faculty of Electrical and Computer Engineering, Technische Universität Dresden. Research interests: model-based systems analysis, system dependability, stochastic models.

KLAUS, Janschek, director of the Institute of Automation at the Faculty of Electrical and Computer Engineering, Technische Universität Dresden, Germany. Research interests: guidance-navigation-control, data fusion, mobile robotics, optical data processing and optomechatronics, model-based systems design.