Titlepage

Abstract page

my dedication...

List of Papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I Andersson, K., Hï¿½mï¿½lï¿½inen, M., Malmqvist, M. (1999) Identification and Optimization of Regeneration Conditions for Affinity-Based Biosensor Assays. A Multivariate Cocktail

Reprints were made with permission from the publishers.

Contents

1. Introduction

SAFETY-CRITICAL EMBEDDED system should be analyzed at the early stages of software development such as *requirements specification*, *software design* and *software allocation* to ensure functional correctness in time, which is more effective and efficient than late analysis [**?**]. Requirements specifications contain the functional and extra-functional requirements that are used in contractual terms and also are used in the subsequent development stages, e.g., design, implementation, verification and validation, etc., hence is vital to maintain good quality of requirements specifications [1][24]. Likewise, software designs should realize the correct functionality of the system at hand before implementation and testing. The software designs are allocated to logical hardware components and units, which are usually constrainted in computation, I/O processing and power provision, thus should also be optimized to accommodate current and future embedded software functionality. In mode-based develpment, the different stages of system development are conceptualized using multiple levels of abstraction as in SysML, AUTOSAR, EAST-ADL, etc. architectural languages.

Over the last decades *formal methods* have attracted the interest of practitioners especially in the safety-critical area to ensure correct functionality at the various levels of system abstraction. Formal methods are mathematical techniques and tools which enable unambiguous specification and modeling, and rigurous analysis of systems, e.g., using theorem proving, model checking, satisfiability-moduo theories (SMT), etc [22]. Their applications have been stagnant mainly due to the difficulty of adapting to the mathematical jargon of the formal languages, lack of tools support, and scalability issues of the methods [2]. As a result, it has become apparent in many cases that complete formal specification and modeling is usually impractical. Rather, formal methods should be (i) used like a “swiss-army-knive”, that is simple, application oriented, mutiple techniques that are of orthogonal; (ii) “lightweight”, that is with partial specification, modeling, and anlayiss on selected safety-critical functionality [11]. In this thesis, we apply formal techniques in this manner to improve quality of requirements specifications and design models, thus we also consider or give focus to scalability and usability of the techniques in order to facilitate their applicability in industry.

Embedded systems requirements are usually expressed in natural language, thus sometimes are ambigous and incomprehensible due to its inherent ambiguity and extreme flexibility, though intuitive and expressive [1]. Template-based specification methods and controlled natural language are the two most commonly used approaches that bases on natural language to improve requirements specifications. The template-based specification methods, e.g., requirements boilerplates [10], etc., lack meta-model for extensibility and the template selection is usually cumbersome. Controlled natural languages, e.g., Attempto [8][7], etc., mimic the intuitiveness of natural language and have formal semantics, however, many lack domain specifity to embedded systems, hence are less effective. In this thesis, we propose the constrained natural language ReSA, which is domain specific to embedded systems and has formal semantics in Boolean logic and description logic. The Boolean representation enable shallow but scalable analysis of requirements, whereas the description logic representation enables deep (or rigorous) analysis but over limited requirements specifications.

The dynamics of safety-critical embedded systems are usually modeled, simulated and anlalyzed before implementation. In this regard, Simulink is one of the most widely used development environment for multi-domain, multi-rate, discrete and continous safety-critical systems in industry. For this main reason, there is an increasing interest in formal analsis of Simulink models. Simulink Design Verifier, which is based on the Prover model-checker, is the de facto tool in the Simulink environment to formally verify Simulink design models. However, it has limited functionality, e.g., it supports only discrete models, has issues with scalability, and lacks verification of timed properties. In this thesis, we propose a less exhaustive, yet rigorous, scalable and robust analysis technique based on the statistical model checking [**?**], to detect functional and complex timing requirements, e.g., end-to-end timings, temporal order of actions, etc. In contrast to the exact model checking techniques, the statistical model checking technique uses adequate execution traces to properties of interest and scales better on the expense of less coverage.

In the presence of limited critical system resources such as processor, memory and power provision of embedded hardwares, safety-critical software with complex functionality, e.g., automotive functions, are distributed on multiple computing units to gain additional power of computation. Moreover, they can share execution platforms along non-critical software to improve efficiency following *mixed-critical* design, which enables non-interference of software applications with different criticality. In this case, maximizing reliability of safety-critical software is crucial to meet high-level reliability goals to improve overall dependability of the system. Fault-tolerance is the most common technique to improve reliability but increases overhead of computation and consumes more power and energy. Therefore, it is crucial to make sure distributed safety-critical software systems are predictable, e.g., meet timing constraints and reliability goals, but also consumes less power to accommodate the ever increasing functionality of embedded software.

In the context of distributed computing, the timing and relibility analysis is not trivial while considering optimization of power cosumption due to the conflicting nature of the properties, e.g., to reduce the total power consumption of a distributed software, less processors (or computing units) are needed, however, to meet end-to-end timing requirements additional computing units may be required, which is also the case if the reliability of the system is need to be maximized such as by applying fault-tolerance in order to meet reliability goals. In this work, we propose an integrated approach to meeting the timing requirements and reliability goals while minimizing the total power consumption of the distributed system using exact and heuristic methods to efficiently allocate safety-critical and non-safety-critical software on network of computing units, with limited computation capability and power specifications.

Our research is evaluated on industrial automotive use cases and realistic benchmark. The requirements specification language ReSA and the Simulink models anlaysis are evalauted on the adjustable speed-limiter (ASL) and brake-by-wire (BBW) systems provided by Volvo Group Trucks Technology (VGTT). ASL is a speed limitation automotive function which controls the vehicle speed of Volvo trucks from speeding up and is useful in roads where speed-limitation signs are in place. The ASL use case consists of arund 300 functional and extra-functional requirements, architctural models in EAST-ADL, and Simulink models. The integrated software allocation is evaluated on the engine management system benchmark provided by Bosch [], which consist of an AUTOSAR architecure with the timing specifications, activation mechanisms of schedulable objects employed to model the execution behavior of the system.

1.1 Research Contributions Overview

In this subsection, we give overview of the thesis contributions, and later in Section x, the contributions are further discussed in detail.

- **Formal Analysis of natural language requirements:** Most software and system requirements of embedded systems are specified in natural language, in fact it has become the de facto standard in industry, which is mainly because it is intuitive as opposed to computer languages, but also is expressive and flexible that many software engineers and other stakeholders find it easy to use. However, it is inherently ambiguous and therefore could lead to ambiguous and incomprehensible specifications. As opposed to the use of templates [10], specification patter systems[9, 13] , we propose a fairly expressive, flexible yet structured and domain domain-specific language, called *ReSA* [ref] that utilizes the EAST-ADL architectural language to improve its effectiveness by reducing syntactic and semantic ambiguities of specifications. The language has translation in Boolean and description logic to support rigorous analysis using existing formal method tools, e.g., SMT solving, reasoners (inference engines) to detect, e.g., logical inconsistencies. Moreover, by translating interesting specifications to TCTL and WMTL properties, the language can be used to abstract syntactic complexity there by simplifying properties specification, e.g., for use in the model checking of Simulink model, after translation to formal model as briefly discussedin the next contribution.

- **Scalable analysis of Simulink models:** Many safety-critical embedded systems are developed in Simulink [12], which is the de facto modeling language employed in industry.To provide assurance that Simulink models fulfill given functional and timing requirements, we propose a pattern-based, execution-order preserving automatic transformation of atomic and composite Simulink blocks into stochastic timed automata that can be formally analyzed using UPPAAL Statistical Model Checker [3]. Our method is scalable, and has been validated on industrial use cases [5]. The statistical model checker analyzes a state-transition system by conducting statistical analysis on the collected traces of the system executions, effectively mitigating the state-space explosion of (exact) model checking [16].

- **Deployment optimization of distributed software applications:** At the system design level, the requirements specifications are realized by a system architecture of software and hardware parts that are constructed by functional components (or modules) that abstract the functionality of the system. In the platform-based development approach [23], the system design should take into account consumption of critical hardware resources, such as power consumption, for two main reasons: optimizing power consumption is beneficial in order i) to accommodate more applications as well as to support power-intensive applications, and ii) to increase battery-life by lowering the amount of heat released by electronic components in the system. In this thesis, we propose an exact software allocation approach, as well as a heuristic one, for multirate systems that need to meet both timing and reliability.

- **Validation on industrial use cases:** Our contributions such as its the ReSA language as well as the proposed formal analysis of Simulink model is validated on industrial use cases, which are provided

The rest of the thesis proposal is organized as follows. Section **??** introduces the research goals and the scientific contributions to address the research goals. Section **??** presents the research methods applied to conduct the research especially in the context of academic-industry collaboration.

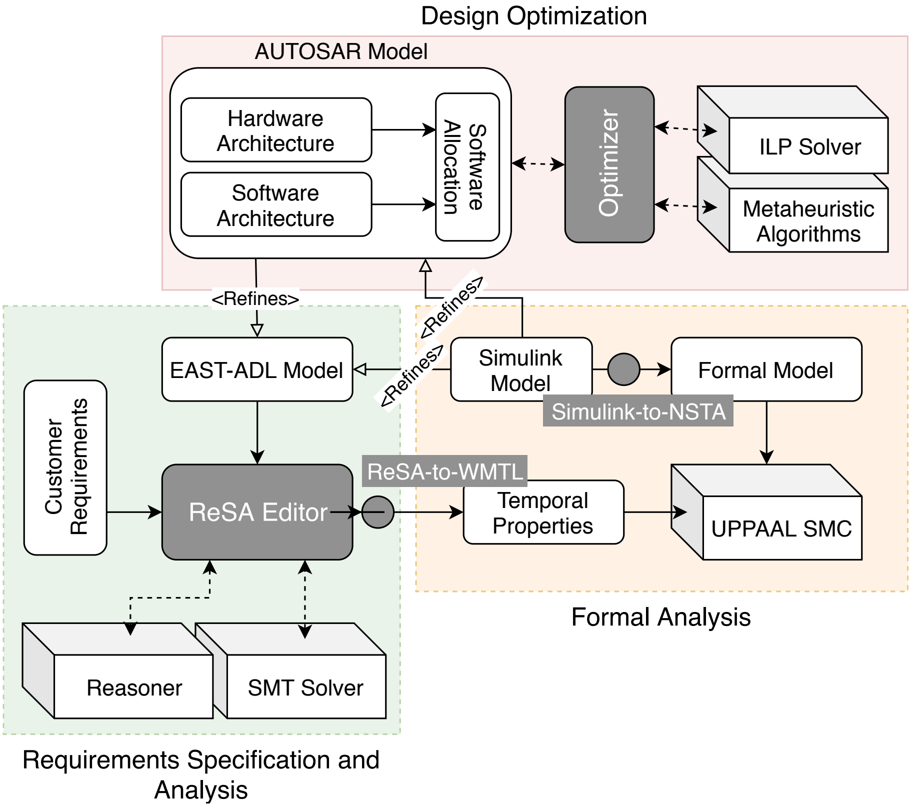

Figure 1.1: Thesis Contributions Workflow.

Section **??** shows the proposed outline of the thesis, followed by the presentation of the research progress in Section **??**, including the time plan until the doctoral defense. Section **??** presents the third-cycle outcomes, adapted from the individual study plan (ISP). Finally, Section **??** discusses the related work before concluding the proposal in Section **??**.

1.2 Thesis Outline Overview

The thesis is divided into two parts. The first part is a summary of our research. It is organized as follows: in Chapter 2, we give the background information on description logic, Boolean satisfiability problem, Simulink, stochastic timed automata, and meta-heauristic optimization. In Chapter 3, we explain the research problem and outline the research goals. The thesis contributions are discussed in Chapter 4, followed by the related work in Chapter 5. In Chapter 3, we describe the research method applied to conduct the research. Finally, in Chapter 7, we conclude the thesis and outline possible directions for future work.

The second part is a collection of papers included in thethesis, and briefly described listed as follows:

2. Preliminary

3. Problem Formulation

The research is motivated by problems encountered in the development of automotive systems from Volvo Group Trucks Technology (VGTT) and Scania, specifically in the area of requirements specification and verification of automotive systems. The problem is articulated under the VeriSpec project[1] with particular focus on effectiveness, scalability and engineer-friendliness of new methods and tools, and seamless integration such tools into existing development platorms.

Based on preliminary study of automotive software development at VGTT through informal discussions, interviews, use cases analysis and first-hand experience on existing automotive systems development tools, we identify three important issues that can potentially impact the dependability of safety-critical automotive systems and the performance of the development process. The isssues are summarized as follows: i) propretary and open-source IDEs are used to develop automotive systems such as SystemWeaver variant, EATOP variant. The IDEs use natural language for the specification of requirements, which are linked to the system models. Neverthless, the specifications are sometimes incomprehensible and inconsistent due to the inherent ambiguity of natural language as well as errors introduced due to human factors, ii) Simulink is frequently used to model, simulate autmotive systems before code generation at VGTT. However, there is lack of early verifiation especially related to timing and functional verification at the model-level, that is before code-generation, iii) finally, we observe increasing complexity of the volvo truck eleccal/electronic system due to advanced functionaly intoduced to assist the driver and improve safety.

The above problems have laid the ground for the research in the Ph.D. thesis. In light of the problems, the overall research goal of the thesis is to:

> Provide assurance of functionality and quality of embedded software systems, at various levels of abstraction, via formal analysis and optimization of critical system resources.

The overall research goal is refined into five research goals in order to simplify addressing the research problems by targeting specific parts

[1] http://www.es.mdh.se/projects/343-VeriSpec___Structured_Specification_and_Automated_Verification_for_Au

3.1 Research Goals

The first research goal addresses the need for improved method of requirements specification in natural language, which is the de facto standard to specify requirements in industry. It is intuitive, expressive, and flexible, hence easy to use. However it is inherently ambiguous and therefore could lead to undesirable (or bad quality specifications), that is incomprehensible and ambiguous. Although template-based specification methods, e.g., requirement boilerplates, specification pattern systems, can reduce ambiguity and also improve readability, they are usually restrictive, and selecting appropriate templates is usually cumbersome. Likewise, existing controlled natural languages do not effectively specify embedded systems requirements due to lack of specialization in domain knowledge, e.g., Attempto, PING, etc. Therefore, the first research goal is to:

> RG 1: Improve the comprehensibility and analyzability of natural language requirements by applying domain-specific knowledge of embedded systems and defining formal semantics to the specifications.

One of the mechanisms to improve natural language specifications is by constraining the language, including its syntax, semantics and the lexicon [14]. The design of a constrained natural language for the specification of requirements is not trivial mainly because: i) natural language possesses conflicting properties (or metrics) [1], for instance by constraining natural language, its expressiveness and flexibility [21] can be impaired. Furthermore, it could lose its intuitiveness. Therefore, appropriate trade-offs should be made during the design in order to have a robust and effective specification language; ii) in order to achieve a usable constrained natural language, the latter should be tailored to a specific domain of application, which requires expertise in the field; iii) usually requirements are specified before the system architecture is developed, so the constrained natural language should provide support to analyze requirements in the absence of architectural models.

Besides assuring the quality of requirements specifications individually, the latter should be analyzed in ensemble in order to detect errors that span multiple specifications, e.g., logical inconsistencies within specifications. In essence, the specifications should possess rich semantics that relate individual specifications to each other. In this regard, the second research goal is to:

> RG 2: Facilitate rigorous analysis of requirements specifications expressed in constrained natural language through transformation to Boolean and description logics.

Natural language requirements specifications are constructed from syntactic units, such as words, phrases, clauses, statements, etc. Consequently, rigorous analysis of specifications involve parsing and interpreting the syntactic units, which is a complex problem in computational linguistics, mainly due to the multitude of interpretation (or semantics) paradigms [4]. Model-theoretic semantics is a widely-applied paradigm in computational semantics. It computes truth values of sentences by inductively applying interpretation functions on the syntactic units (or structures) of the sentences in relation to mathematical systems, e.g., propositional, first-order systems. The depth of the interpretation greatly affects the use and applicability of the interpretation approach. For instance, with the assumption that a proposition equates to a clause in a sentence, propositional logic elegantly represents sentences and scales well to find truth values. However it provides a shallow interpretation of the sentences as it abstracts the various units of clauses by propositional variables. In contrast, first-order logic representations provide richer semantics and thus enable rigorous anlaysis, but yet lesser tractable. Therefore, the type of analysis and level of rigor needed drives the appropriate semantics definition.

The task of verification and validation is vital in the development of safety-critical systems. Basically, for correct functionality, verifying the timing behavior of safety-critical systems is as important as their functional behavior counterpart which include checking for time-bounded response of actions, timing constraints on sequence of actions, etc. In this thesis, we focus on systems developed using Simulink as it is among (if not the most) widely-used model-based design, analysis, simulation and code-generation environment for developoing safety-critical and real-time embedded system. It is frequently used at VGTT and Scania, as well as in similar embedded systems industries, e.g., automation, avionics, etc. In this thesis, we target Simulink models that are hybrid (that is, use continuous and discrete Simulink blocks), and contains Simulink blocks that are triggered using different sampling rates. The latter cases causes under sampling and over sampling effects of signals that make timing analysis challenging.

Existing Simulink analysis techniques, e.g., by type checking, simulation, and formal verification via Simulink Design Verifier (based Prover model-checker) (SDV[2]) are not sufficient to address the full correctness of safety-critical real-time Simulink models. The latter lacks support for checking temporal correctness as specified in timed properties, e.g., in TCTL, and also lacks support for verifying continuous models and suffers from scalability due to its reliance on exact model-checking [17]. In contrast to exact model checking, statistical model-checking collects sufficient traces of system simulations, and consequently applies statistical methods to verify properties. It is found to be scalable and its accuracy be improved by taking many traces of simualtions. In this regard, the fourth research goal is to:

[2] https://se.mathworks.com/products/sldesignverifier.html

RG 3: Enable scalable formal analysis of multirate and hybrid Simulink models using statistical model-checking.

Simulink consists of connected and hierarchical Simulink blocks, which encode mathematical functions. For industrial systems, the number of blocks in a Simulink model can be in the order of thousands, and the blocks can be triggered with different sampling frequencies for discrete blocks and without any sampling frequency for continuous blocks. Therefore, typical industrial Simulink models are usually complex and comprise mixed signals, multiple rates, discrete and continues Simulink blocks, making formal analysis challenging.

Afterwards, the specifications are used to create the high-level system design (or architecture) that realizes the required functionality. The system architecture consists of software and hardware logical components, as well as mappings from the software components to the hardware components (or computation nodes). The latter activity of the system design, also known as *software allocation*, should be handled effectively in order to preserve the functionality of the software architecture, including functional correctness and extra-functional attributes such as timing and reliability. Furthermore, the allocation should be efficient in order to conserve critical system resources while satisfying requirements as well as design and hardware constraints.

Effective and efficient allocation of multirate software applications is highly needed, as it is crucial to safety-critical systems such as automotive systems, aircrafts, etc. Many multirate applications are deployed on several computation nodes, which are possibly heterogeneous, over shared communication networks. Therefore, the third research goal is as follows:

RG 4: Find an effective and efficient allocation of multirate software applications on a network of heterogeneous computing nodes, with respect to critical system resources.

Allocation of multirate software applications on a network of computation nodes is challenging mainly due to the complex timing analysis that needs to be considered during the allocation process [19, 20]. The timed paths of multirate applications increasing the search space, and therefore finding an efficient allocation becomes exponential. Furthermore, the heterogeneity of the computation nodes forces the search method to consider every computation node for better result, thus increasing the optimization time, as opposed to considering homogeneous nodes. Since the software allocation could be intractable especially for large systems, methods based on heuristics and approximation should be provided instead of exact ones.

Addressing research goal 3 ensures that the software applications are deployed with minimal resource utilization at the same time assuring the satisfaction of extra-functional requirements such as timing and reliability. Furthermore, it ensures that the design and hardware constraints are met.

The system architecture, in particular the software architecture is further refined via software unit designs that capture the behavior of the software. We assume that the latter is captured by models in Simulink [12], which is one of the most popular and robust component-based software development environment employed in industry.

In order to show the validity of our proposed solutions, a working prototype should be developed and should be applied on industrial uses cases. Our proposed specification and analysis methods and tools should be engineering friendly in order to facilitate their adoption in industry, for instance they should embody intuitiveness and seamless integration into industrial practices. Therefore, the last research goal is as follows:

RG 5: Provide automated and engineering-friendly support for the requirements specification, software allocation of embedded and formal analysis of Simulink models.

Seamless integration of new development methods and tools require appropriate interfaces to plug into existing development methods and tools in order to facilitate adoption of formal techniques, by lowering the required effort and cost. In particular, formal methods should be accompanied by engineering-friendly interfaces, as the syntax of the formal notations is not familiar to most engineers, likewise, understanding the underlying semantics requires a substantial shift from traditional software development paradigms. In order to tackle these challenges, very close cooperation with engineers and know-how of domain-specific industrial tools and practices are paramount.

3.2 Papers Included in the Thesis

The papers to be included in the PhD thesis are listed as follows.

3.2.1 Paper A

ReSA: An Ontology-based Requirement Specification Language Tailored to Automotive Systems

Nesredin Mahmud, Cristina Seceleanu, Ljungkrantz Oscar

Abstract: *Automotive systems are developed using multi-leveled architectural abstractions in an attempt to manage the increasing complexity and criticality of automotive functions. Consequently, well-structured and unambiguously specified requirements are needed on all levels of abstraction, in order to enable early detection of possible design errors. However, automotive industry often relies on requirements specified in ambiguous natural language, sometimes in large and incomprehensible documents. Semi-formal requirements specification approaches (e.g., requirement boilerplates, pattern-based specifications, etc.) aim to reduce requirements ambiguity, without altering their readability and expressiveness. Nevertheless, such approaches do not offer support for specifying requirements in terms of multi-leveled architectural concepts, nor do they provide means for early-stage rigorous analysis of the specified requirements. In this paper, we propose a language, called ReSA, which allows requirements specification at various levels of abstraction, modeled in the architectural language of EAST-ADL. ReSA uses an automotive systems' ontology that offers typing and syntactic axioms for the specification. Besides enforcing structure and more rigor in specifying requirements, our approach enables checking refinement as well as consistency of requirements, by proving ordinary boolean implications. To illustrate ReSA's applicability, we show how to specify some requirements of the Adjustable Speed Limiter, which is a complex, safety-critical Volvo Trucks user function.*

Status: Published in *10th IEEE International Symposium on Industrial Embedded Systems (SIES), 2015, IEEE*

My Contribution: I was the main driver of the paper. I developed the ReSA language including its syntax and semantics, and Cristina Seceleanu proposed a consistency analysis technique besides giving useful comments and ideas on the design of the language. Oscar Ljungkrantz provided useful materials from VGTT that were eventually analyzed for the language development, and gave feedback on the language design and implementation from an industrial viewpoint.

3.2.2 Paper B

ReSA Tool: Structured Requirements Specification and SAT-based Consistency-checking

Nesredin Mahmud, Cristina Seceleanu, Ljungkrantz Oscar

Abstract: *Most industrial embedded systems requirements are specified in natural language, hence they can sometimes be ambiguous and error-prone. Moreover, employing an early-stage model-based incremental system development using multiple levels of abstraction, for instance via architectural languages such as EAST-ADL, calls for different granularity requirements specifications described with abstraction-specific concepts that reflect the respective abstraction level effectively. In this paper, we propose a toolchain for structured requirements specification in the ReSA language, which scales to multiple EAST-ADL levels of abstraction. Furthermore, we introduce a consistency function that is seamlessly integrated into the specification toolchain, for the automatic analysis of requirements logical consistency prior to their temporal logic formalization for full formal verification. The consistency check subsumes two parts: (i) transforming ReSA requirements specification into boolean expressions, and (ii) checking the consistency of the resulting boolean expressions by solving the satisfiability of their conjunction with the Z3 SMT solver. For validation, we apply the ReSA toolchain on an industrial vehicle speed control system, namely the Adjustable Speed Limiter.*

Status: Published in *Federated Conference on Computer Science and Information Systems (FedCSIS), 2016, IEEE*

My Contribution: I was the main driver of the paper. I developed the ReSA toolchain that consists of the editor and the consistency checker including the integration with the Z3 SAT solver in the backend. Cristina Seceleanu formulated the consistency checking and together with Oscar Ljungkrantz, they contributed to the paper with useful comments and ideas.

3.2.3 Paper C

Specification and Semantic Analysis of Embedded Systems Requirements: From Description Logic to Temporal Logic

Nesredin Mahmud, Cristina Seceleanu, Ljungkrantz Oscar

Abstract: *Due to the increasing complexity of embedded systems, early detection of software/hardware errors has become desirable. In this context, effective yet flexible specification methods that support rigorous analysis of embedded systems requirements are needed. Current specification methods such as pattern-based, boilerplates normally lack meta-models for extensibility and flexibility. In contrast, formal specification languages, like temporal logic, Z, etc., enable rigorous analysis, however, they usually are too mathematical and difficult to comprehend by average software engineers. In this paper, we propose a specification representation of requirements, which considers thematic roles and domain knowledge, enabling deep semantic analysis. The specification is complemented by our constrained natural language specification framework, ReSA, which acts as the interface to the representation. The representation that we propose is encoded in description logic, which is a decidable and computationally-tractable ontology language. By employing the ontology reasoner, Hermit, we check for consistency and completeness of requirements. Moreover, we propose an automatic transformation of the ontology-based specifications into Timed Computation Tree Logic formulas, to be used further in model checking embedded systems.*

Status: *Published in 15th International Conference Software Engineering and Formal Methods (SEFM), 2017, LNCS Springer.*

My Contribution: I was the main driver of the language. I developed the ReSA language semantics using event-base approach, which is encoded in description logic. Cristina Seceleanu and Ljungkrantz Oscar provided with useful ideas and comments.

3.2.4 Paper D

Scalable Allocation of Fault-tolerant Multi-rate AUTOSAR Applications

Nesredin Mahmud, Guillermo Rodriguez-Navas, Hamid Faragardi,Saad Mubeen, Cristina Seceleanu

Abstract: *Software-to-hardware allocation plays an important role in the development of resource-constrained automotive embedded systems that are required to meet timing, reliability and power requirements. This paper proposes an Integer Linear Programming optimization approach for the allocation of fault-tolerant embedded software applications that are developed using the AUTOSAR standard. The allocation takes into account the timing and reliability requirements of the multi-rate cause-effect chains in these applications and the heterogeneity of their execution platforms. The optimization objective is to minimize the total power consumption of the applications that are distributed over more than one computing unit. The proposed approach is evaluated using a range of different software applications from the automotive domain, which are generated using the real world automotive benchmark. The evaluation results indicate that the proposed allocation approach is effective and scalable while meeting the timing, reliability and power requirements in small- and medium-sized automotive software applications.*

Status: To be submitted to *Journal of Systems and Software (JSS), Elsevier.*

My Contribution: I am the main driver of the paper. I developed the system model (including the power consumption, timing, reliability models) and further refined by the co-authors. Hamid Faragardi and I developed the ILP model, and I implemented the ILP problem (including the system model) and collected experimental results. The co-authors gave useful ideas and comments on respected parts of the paper: Guillermo Rodriguez-Navas on reliability modeling, Hamid Faragardi on optimization and related work, Saad Mubeen on the timing analysis, and Cristina Seceleanu gave comments and ideas on the main contributions of the paper, including on the optimization objective and constraints.

In Case of Rejection: Power-aware Allocation of Fault-tolerant Multi-rate AUTOSAR Applications conference paper

Status: To appear in Proc. of the *25th Asia-Pacific Software Engineering Conference (APSEC), 2018, Japan*, IEEE CS.

Note: The JSS paper is an extension of the APSEC paper.

3.2.5 Paper E

SIMPPAAL - A Framework For Statistical Model Checking of Industrial Simulink Models

Predrag Filipovikj, Nesredin Mahmud, Raluca Marinescu, Cristina Seceleanu, Oscar Ljungkrantz, Henrik Lönn

Abstract: *The evolution of automotive systems has been rapid. Nowadays, electronic brains control dozens of functions in vehicles, like braking, cruising, etc. Model-based design approaches, in environments such as MATLAB Simulink, seem to help in addressing the ever-increasing need to enhance quality, and manage complexity, by supporting functional design from a set of block libraries, which can be simulated and analyzed for hidden errors, but also used for code generation. For this reason, providing assurance that Simulink models fulfill given functional and timing requirements is desirable. In this paper, we propose a pattern-based, execution-order preserving automatic transformation of atomic and composite Simulink blocks into stochastic timed automata that can then be formally analyzed with Uppaal Statistical Model Checker (Uppaal SMC). To enable this, we first define the formal syntax and semantics of Simulink blocks and their composition, and show that the transformation is provably correct for a certain class of Simulink models. Our method is supported by the SIMPPAAL tool, which we introduce and apply on two industrial Simulink models, a prototype called the Brake-by-Wire and an operational Adjustable Speed Limiter system. This work enables the formal analysis of industrial Simulink models, by automatically generating stochastic timed automata counterparts.*

Status: To be submitted (by 15th October 2018) to the ACM Transactions on Software Engineering and Methodology (TOSEM) Journal, ACM.

My Contribution: The three co-authors contributed equally to writing the paper. Technically, I equally contributed with proposing the pattern-based semantics of Simulink blocks, together with Predrag Filipovikj. I introduced a mechanism to enforce the execution order of the blocks using inter-arrival times. Predrag implemented the flattening algorithm and the tool for the automatic transformation of Simulink models into a network of timed automata with stochastic semantics. Raluca Marinescu contributed with analyzing the BBW system, Cristina Seceleanu contributed with defining the methodology, and with useful ideas and comments. Guillermo Rodriguez-Navas wrote the related work section. The industrial coauthors provided the use cases and commented on the final draft.

In Case of Rejection: It will be included as a technical report [6].

3.2.6 Paper F

SIMPPAAL meets ReSA: From Automated Requirements Specifications to Automated Formal Analysis of Simulink Models.

[authors part]

Abstract: *The main objective of this work is to extend the validation of our contributions, that is the SIMPPAAL method and tool [6] for the verification of Simulink models based on SMC, for which the properties will be automatically generated by employing the structured requirements specification framework called ReSA [18]. The verification will be applied on a set of Simulink models that are part of the Adjustable Speed Limiter system, which is an operational system installed in all Volvo Trucks.*

Status: Candidate venue for submission: *"NFM 2019: 11th Annual NASA Formal Methods Symposium"* Submission deadline: 2018-12-14. Notification Date: 2019-02-22.

In Case of Rejection: SEKE 2019: International Conference on Software Engineering and Knowledge Engineering. Submission Deadline: 2019-03-01. Notification date: 2019-04-20.

Last but not least we show how to transform structured requirements into temporal logics for exhaustive and statistical verification of models, and integrate the developed tool support for requirements specification and analysis, and model checking of Simulink models into the VeriSpec framework.

4. Contributions

5. Research Method

6. Related Work

7. Conclusions and Future Work

Bibliography

[1] ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering. *ISO/IEC/IEEE 29148:2011(E)*, pages 1–94, 12 2011.

[2] Jean-Raymond Abrial. Formal Methods in Industry: Achievements, Problems, Future. *Proceedings of the 28th International Conference on Software Engineering*, 2006.

[3] Peter Bulychev, Alexandre David, Kim Gulstrand Larsen, Marius Mikučionis, Danny Bøgsted Poulsen, Axel Legay, and Zheng Wang. UPPAAL-SMC: Statistical Model Checking for Priced Timed Automata. *Electronic Proceedings in Theoretical Computer Science*, 2012.

[4] Alexander Clark, Chris Fox, and Shalom Lappin. *The Handbook of Computational Linguistics and Natural Language Processing*. 2010.

[5] P. Filipovikj, N. Mahmud, R. Marinescu, C. Seceleanu, O. Ljungkrantz, and H. Lönn. *Simulink to UPPAAL statistical model checker: Analyzing automotive industrial systems*, volume 9995 LNCS. 2016.

[6] Predrag Filipovikj, Nesredin Mahmud, Raluca Marinescu, Guillermo Rodriguez-Navas, Cristina Seceleanu, Oscar Ljungkrantz, and Henrik Lönn. Analyzing Industrial Simulink Models by Statistical Model Checking. Technical report, 3 2017.

[7] Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto Controlled English for Knowledge Representation. pages 104–124. Springer, Berlin, Heidelberg, 2008.

[8] Norbert E Fuchs and Rolf Schwitter. Attempto Controlled English {(ACE)}. *CoRR*, cmp-lg/960, 1996.

[9] Volker Gruhn and Ralf Laue. Patterns for Timed Property Specifications. *Electronic Notes in Theoretical Computer Science*, 2006.

[10] Elizabeth Hull, Ken Jackson, and Jeremy Dick. *Requirements Engineering*. Springer London, London, 2011.

[11] Daniel Jackson. Lightweight Formal Methods. In JosÃ© Nuno Oliveira and Pamela Zave, editors, *FME 2001: Formal Methods for Increasing Software Productivity*, page 1, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[12] Thomas L. Harman James B. Dabney. *Mastering Simulink*. Pearson, 2003.

[13] Sascha Konrad and Betty H. C. Cheng. Real-time specification patterns. In *Proceedings of the 27th international conference on Software engineering - ICSE '05*, page 372, New York, New York, USA, 2005. ACM Press.

[14] Tobias Kuhn. A Survey and Classification of Controlled Natural Languages. *Computational Linguistics*, 40(1):121–170, 3 2014.

[15] Edward Ashford Lee and Sanjit Arunkumarr Seshia. *Introduction to Embedded Systems: A Cyber-Physical Systems Approach.* 2011.

[16] Axel Legay, BenoÃ®t Delahaye, and Saddek Bensalem. Statistical Model Checking: An Overview. pages 122–135. Springer, Berlin, Heidelberg, 2010.

[17] Florian Leitner and Stefan Leue. Simulink Design Verifier vs. SPIN a Comparative Case Study. *Proceedings of FMICS*, 2008.

[18] N Mahmud, C Seceleanu, and O Ljungkrantz. ReSA Tool: Structured requirements specification and SAT-based consistency-checking. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1737–1746, 9 2016.

[19] Nesredin Mahmud, Guillermo Rodriguez-Navas, Hamid Reza Faragardi, Saad Mubeen, and Cristina Seceleanu. Power-aware Allocation of Fault-tolerant Multi-rate AUTOSAR Applications. In *25th Asia-Pacific Software Engineering Conference*, 12 2018.

[20] Saad Mubeen, Jukka Mäki-Turja, and Mikael Sjödin. Support for End-to-end Response-time and Delay Analysis in the Industrial Tool Suite: Issues, Experiences and A Case Study. *Computer Science and Information Systems*, 10(1):453–482, 2013.

[21] Andriy Myachykov, Christoph Scheepers, Simon Garrod, Dominic Thompson, and Olga Fedorova. Syntactic flexibility and competition in sentence production: The case of English and Russian. *Quarterly Journal of Experimental Psychology*, 2013.

[22] Gerard OâRegan. Concise guide to formal methods, 2017.

[23] Alberto Sangiovanni-Vincentelli, Luca Carloni, Fernando De Bernardinis, and Marco Sgroi. Benefits and Challenges for Platform-based Design. In *Proceedings of the 41st annual conference on Design automation - DAC '04*, page 409, New York, USA, 2004. ACM Press.

[24] FÃ¡bio Levy Siqueira. Comparing the comprehensibility of requirements models: An experiment replication. *Information and Software Technology*, 96:1–13, 4 2018.