Description of My Previous Work

Holger Giese
Software Engineering Group, University of Paderborn
Warburger Str. 100, D-33098 Paderborn, Germany
hg@uni-paderborn.de

Introduction

Today, advanced technological products often employ information and communication technology to enhance their functionality and quality. Therefore, nowadays software plays an important role in the control of many technical systems such as electrical and mechanical devices, technical processes in the chemical industry, or a modern automobile. In the future, software-intensive systems are in addition expected to be highly distributed and to exhibit adaptive and anticipatory behavior when operating in highly dynamic environments and interfacing with the physical world. Therefore, appropriate modeling and analysis techniques to address these software-intensive systems are required that support a mix of models from a multitude of disciplines such as software engineering, control engineering, and business process engineering.

Previous Work

The identified challenges for self-adaptive software-intensive systems have been addressed in my research from a software engineering and distributed system perspective. The background of these efforts is the development of techniques which enable to systematically study architectural decisions and detect problems early on. The techniques include suitable visual modeling techniques for the software architecture of distributed systems [1][2], the modeling of service- and component-based systems [3], and an approach for the consistent integration of different architectural views by means of synthesis [4].

Software-intensive systems are often not only distributed but also contain embedded systems which must exhibit hard real-time behavior. To also address these requirements, an approach to model such software-intensive systems and their architectures a multi-agent system with autonomous, self-adaptive agents using UML components and patterns with real-time behavior [5][6] has been developed. In addition, we made first steps towards the synthesis of the real-time behavior from scenarios [7] to ease the transition from requirements to design.

To be able to integrate the models from the other disciplines usually involved in the development of software-intensive systems, such as control engineering and business process engineering, into our software engineering model, we studied the integration of workflow and UML models [8] and the integration of control-oriented quasi-continuous models into hierarchical UML components and Statecharts [9]. The concepts presented in [9] can be further employed to also describe self-adaptive behavior in form of hierarchical reconfiguration and support the decoupling of soft and non real-time task at the architectural level.

As software-intensive systems are often safety-critical systems, suitable means to identify and analyze relevant hazards are mandatory. A compositional hazard analysis which exploits the component structure of the system [10] and an approach for the automatic compositional verification of embedded real-time systems using components and patterns [6] and the modular verification of the hierarchical hybrid systems with modular reconfiguration capabilities [11] have thus been developed. The verification approach is supplemented by a technique to verify structural safety properties for infinite state systems with structural adaptation behavior at run-time [12].

We further developed concepts for the code-generation of hard real-time systems [13] which ensure that the high level properties of the models which have been analyzed and verified can also be guaranteed for the generated code.

The results obtained for the modeling, analysis, synthesis, and code generation are supported by the Fujaba Real-Time Tool Suite. This real-time version of the UML CASE tool Fujaba Tool Suite provides seamless tool support for the model-driven development of self-optimizing mechatronic systems [14].

In the context of the model-driven development of software, some general concepts for the integration of different modeling artifacts at the meta-model level have been developed [15].

Planned Short Presentation

Model-Driven Development of Dependable Mechatronic Multi-Agent Systems

Advanced mechatronic systems of the future are expected to behave more intelligent than today by building communities of autonomous agents which exploit local and global networking to enhance their behavior and to realize otherwise not possible functionality. While engineering of mechatronic systems and software engineering for embedded systems, multi-agent systems, and distributed systems are established areas, no solution for the systematic development of the outlined future generation of intelligent, distributed, embedded systems exists today. This problem is not simply a matter of composing the solutions developed for each of these areas as some of their requirements are in conflict: E.g., flexibility and autonomy are to some extent at odds with predictability and safety. In addition, the gap between the discrete software world and the continuous world of control theory has to be bridged to enable the full potential of such self-adaptive and self-optimizing mechatronic systems.

Within the Collaborative Research Centre 614 in Paderborn, a model-driven development approach is under development to address this challenge. A restricted high level UML modeling approach coined Mechatronic UML with specific extensions for the modeling of real-time behavior and hybrid behavior serves as a basis for rigorous verification to address the correctness and safety issues. Within Mechatronic UML, components and coordination patterns serve as a basis for the compositional verification of the real-time coordination of systems with dynamically changing structure. Hybrid components and their embedding in the discrete real-time coordination by means of Hybrid Reconfiguration Charts further support the design and verification of elaborated multi-level reconfiguration for whole hierarchies of mechatronic assemblies. These analysis results for the high level models are preserved by also developed code synthesis capabilities and run-time support, which guarantee by construction that all properties of the high level models also hold for the final implementation.

Short CV

Holger Giese is since 2001 assistant professor for object-oriented specification of distributed systems in the Software Engineering Group of the University of Paderborn. He studied technical computer science at the University Siegen and received his engineering degree in October 1995. He received a doctorate in Computer Science at the Institute of Computer Science at the University of Mnster in February 2001.

His research interests are software engineering for distributed, component-based and real-time systems. This includes techniques and tools for the modeling, analysis, validation, and verification of safety-critical distributed real-time systems with mechatronic components using design patterns, components, and software architecture modeling.

He is a member of the collaborative research center for Self-optimizing Concepts and Structures in Mechanical Engineering and one of the directors of the B1 subproject for software design techniques.

He is member of the Association for Computing Machinery (ACM), the IEEE Computer Society, the German Informatics Society (GI), and the German University Council (DHV).

References

[1] Holger Giese and Guido Wirtz. The OCoN Approach for Object-Oriented Distributed Software Systems Modeling. *Computer Systems Science & Engineering*, 16(3):157–172, May 2001.

[2] Holger Giese and Guido Wirtz. Visual Modeling of Object-oriented Distributed Systems. *Journal of Visual Languages and Computing*, 12(2):183–202, April 2001.

[3] Matthias Tichy and Holger Giese. A Self-Optimizing Run-Time Architecture for Configurable Dependability of Services. In Rogério de Lemos, Cristina Gacek, and Alexander Romanovsky, editors, *Architecting Dependable Systems II*, volume 3069 of *Lecture Notes in Computer Science*, pages 25–51. Springer Verlag, 2004.

[4] Holger Giese and A. Vilbig. Separation of Non-Orthogonal Concerns in Software Architecture and Design. *Software and System Modeling (SoSyM)*, 2006. (to appear).

[5] Florian Klein and Holger Giese. Separation of concerns for mechatronic multi-agent systems through dynamic communities. In Ricardo Choren, Alessandro Garcia, Carlos Lucena, and Alexander Romanovsky, editors, *Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications*, volume 3390 of *Lecture Notes in Computer Science*, pages 272–289. Springer Verlag, February 2005.

[6] Holger Giese, Matthias Tichy, Sven Burmester, Wilhelm Schäfer, and Stephan Flake. Towards the Compositional Verification of Real-Time UML Designs. In *Proc. of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering (ESEC/FSE-11)*, pages 38–47. ACM Press, September 2003.

[7] Holger Giese, Florian Klein, and Sven Burmester. Pattern Synthesis from Multiple Scenarios for Parameterized Real-Timed UML Models. In Stefan Leue and Tarja Systä, editors, *Scenarios: Models, Algorithms and Tools*, volume 3466 of *Lecture Notes in Computer Science*, pages 193–211. Springer Verlag, April 2005.

[8] Guido Wirtz, Mathias Weske, and Holger Giese. The OCoN Approach to Workflow Modeling in Object-Oriented Systems. *Information Systems Frontiers*, 3(3):357–376, September 2001.

[9] Sven Burmester, Holger Giese, and Oliver Oberschelp. Hybrid UML Components for the Design of Complex Self-optimizing Mechatronic Systems. In *Informatics in Control, Automation and Robotics*. Kluwer Academic Publishers, 2006. to appear.

[10] Holger Giese, Matthias Tichy, and Daniela Schilling. Compositional Hazard Analysis of UML Components and Deployment Models. In *Proc. of the 23rd International Conference on Computer Safety, Reliability and Security (SAFECOMP), Potsdam, Germany*, volume 3219 of *Lecture Notes in Computer Science*. Springer Verlag, September 2004.

[11] Holger Giese, Sven Burmester, Wilhelm Schäfer, and Oliver Oberschelp. Modular Design and Verification of Component-Based Mechatronic Systems with Online-Reconfiguration. In *Proc. of 12th ACM SIGSOFT Foundations of Software Engineering 2004 (FSE 2004), Newport Beach, USA*, pages 179–188. ACM Press, November 2004.

[12] Holger Giese, Daniela Schilling, Florian Klein, Basil Becker, and Dirk Beyer. Symbolic Invariant Verification for Systems with Dynamic Structural Adaptation. In *Proc. of the* 28^{th} *International Conference on Software Engineering (ICSE), Shanghai, China*. ACM Press, 2006. (accepted).

[13] Sven Burmester, Holger Giese, and Wilhelm Schäfer. Model-Driven Architecture for Hard Real-Time Systems: From Platform Independent Models to Code. In *Proc. of the European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA'05), Nürnberg, Germany*, Lecture Notes in Computer Science, pages 1–15. Springer Verlag, November 2005.

[14] Sven Burmester, Holger Giese, Eckehard Mnch, Oliver Oberschelp, Florian Klein, and Peter Scheideler. Tool Support for the Design of Self-Optimizing Mechatronic Multi-Agent Systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 2006. (accepted).

[15] Sven Burmester, Holger Giese, Jörg Niere, Matthias Tichy, Jörg P. Wadsack, Robert Wagner, Lothar Wendehals, and Albert Zündorf. Tool Integration at the Meta-Model Level within the FUJABA Tool Suite. *International Journal on Software Tools for Technology Transfer (STTT)*, 6(3):203–218, August 2004.