Software Construction

Staff

- Faculty:

  Univ.-Prof. Dr. rer. nat. Horst Lichter
  lichter@swc.rwth-aachen.de

- Secretary:

  Bärbel Kronewetter
  Phone: +49 241 80 21 330
  Fax: +49 241 80 22 352

- Research Assistants:

  Dipl.-Inform. Andreas Ganser
  (third-party funds position)

  Dipl.-Inform. Veit Hoffmann

  Dipl.-Inform. Alexander Nyßen
  (third-party funds position)

  Malek Obaid, M.Sc.
  (DAAD scholarship)

  Chayakorn Piyabunditkul M.Sc.
  (DAAD/NECTEC scholarship)

  Dipl.-Inform. Holger Schackmann
  (third-party funds position)

- Student Researchers:

  Philip Ritzkopf

  Oscar Mendoza

Overview

Our research focuses on the development of new and advanced methods, tools and techniques in the broad area of software construction. Thereby we always try to develop and deliver software engineering support that is applicable under industrial software development conditions.

Currently we are running a couple of research projects (details can be found in the corresponding section), most of them in close cooperation with industrial partners.

In the reporting period we started three new projects. The first aims to combine model driven development and service oriented architecture in the domain of life cycle software. In the second project we try to develop a use case based approach to early simulate functional requirements and to generate test suites. The third is in the area of process improvement where we define a light weight software process improvement approach that has to be especially applicable for small and medium software houses.

Furthermore we have successfully initiated the BugzillaMetrics open source project (hosted by sourceforge). BugzillaMetrics is a metrics tool that was developed in our group. Based on the experience gained with the first version of MeDUSA, the second edition of MeDUSA was released in 2008. Finally, in 2007 DASMA awarded a diploma thesis prize to Lars Grammel, one of our Diploma students, and a Design Pattern intense course was given at the University of Tartu, Estonia.

Research Projects

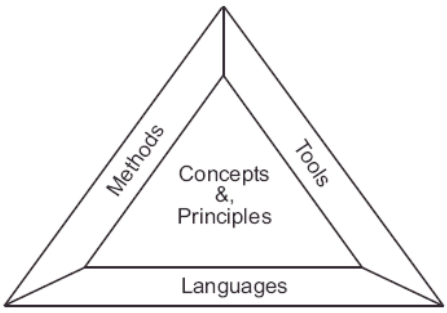
Model-Based Engineering of Small Embedded & Real-Time Systems

Nyßen, H. Lichter

External cooperation: ABB Corporate Research Center, Ladenburg, Germany

As the history of software engineering unveils, abstraction is the means to face complexity. While the 1970's and 1980's have been strongly based on functional and data abstraction, and while the 1990's and early 2000's have been dominated by object-orientation, model-based software development (MBSD) seems to be the predominant development paradigm of the present and upcoming decade.

Due to its increased capabilities in terms of abstraction, traceability and analyzability, it seems to be the in particular useful to address the very special constraints, being faced in the domain of embedded & real-time systems. However, as current surveys unveil, from a state-of-the-practice viewpoint, model-based software development does not seem to have penetrated the embedded & real-time domain yet. Tracing this back to domain-specific technical, economical and organizational constraints, not being properly reflected by current model-based approaches, it is the goal of this project to introduce modern model-based technology and techniques, being however constraint-adequate.



As only a common, integrated methodology, formed by a systematic and concise method, by profound supporting tools, and by underlying languages, being related via common concepts and principles, allows to unleash the full potential of model-based development, it is the central goal of this project to deliver the latter. Regarding the very special constraints, being additionally faced in the domain of small embedded & real-time systems, the approach should in particular be applicable to a domain, which has been very much elided so far.

Model Driven Architectures for Service Oriented Architectures

A. Ganser, H. Lichter

External cooperation: Osthus GmbH, Aachen

Today, profitability and continuous adaptability with respect to business processes and corporate structure are major requirements to companies. Economics claim these goals are achievable only with corporate structure and IT structure collaborating tightly.

But, adjusting structures of companies and their IT is challenging since both exist independently and structures of IT are inflexible in general. Therefore, IT structures inevitably need to change towards well defined components with well defined interfaces. Many approaches have been developed and deployed in order to make IT better foster business processes. But these approaches were often tightened to technical details too much and left organizational aspects out of sight.

Service oriented architectures (short: SOA) are perceived as the silver bullets in that area our days. This is because SOA is an idea which regards business and IT likewise and hence fosters flexibility of the entire company. What's more, it demands loose coupling of components, here called services, and consequently it allows easy exchange of every service.

The IT related sub-discipline of SOA addresses software construction issues and belonging challenges with respect to design, processes, and evolution. These challenges emerge with more impact whenever legacy systems are involved. But SOA allows such systems conceptual and allows to integrate them since they rarely can be build from scratch again. But in most cases "software assets" do not adhere to every requirement of the new environment or the new environment simply adds requirements. Hence, SOA needs to deal with heterogeneous environments on the one hand and on the other hand it needs to tackle subsequent problems.

In the last few months we started our cooperation with Osthus GmbH and picked a topic for initial work in the area of SOA. We aimed for integration of legacy services into state of the art platforms by enhancing these services with extra functionality. Therefore we analyzed the environment thoroughly and built a conceptual model how integration and extension of these services might work. Though the conceptual model is intended to be generic, the guiding motivation is to add security functions to existing services. We did so because there are often existing services that have to be extended with some functionality leaving the source code untouched. This might be due to changes in the environment towards SOA. Finally, we came up with tool support with witch designer can arrange services and assign additional services by simply dragging and dropping existing services from repositories to a design black board.

But this tool support is rather rudimentary for now. This means there are a lot of things designers need to keep in mind during work. Therefore, we aim for better tool support how to warn designer when something is likely to go wrong or when components do not fit together. But the reasons why services might not work together are manifold. Hence, we need to find and extract best practices and condense them to patterns how designing is supposed to be done. In the long run the extracted patterns are intended to be used in model driven development to support model driven architectures.

Methodical and Tool Support for Advanced Use Case Modelling

V. Hoffmann, H. Lichter

Use cases are a widely accepted technique for the elicitation and specification of functional requirements. In practice Use cases often consist of tow parts: an overview diagram which

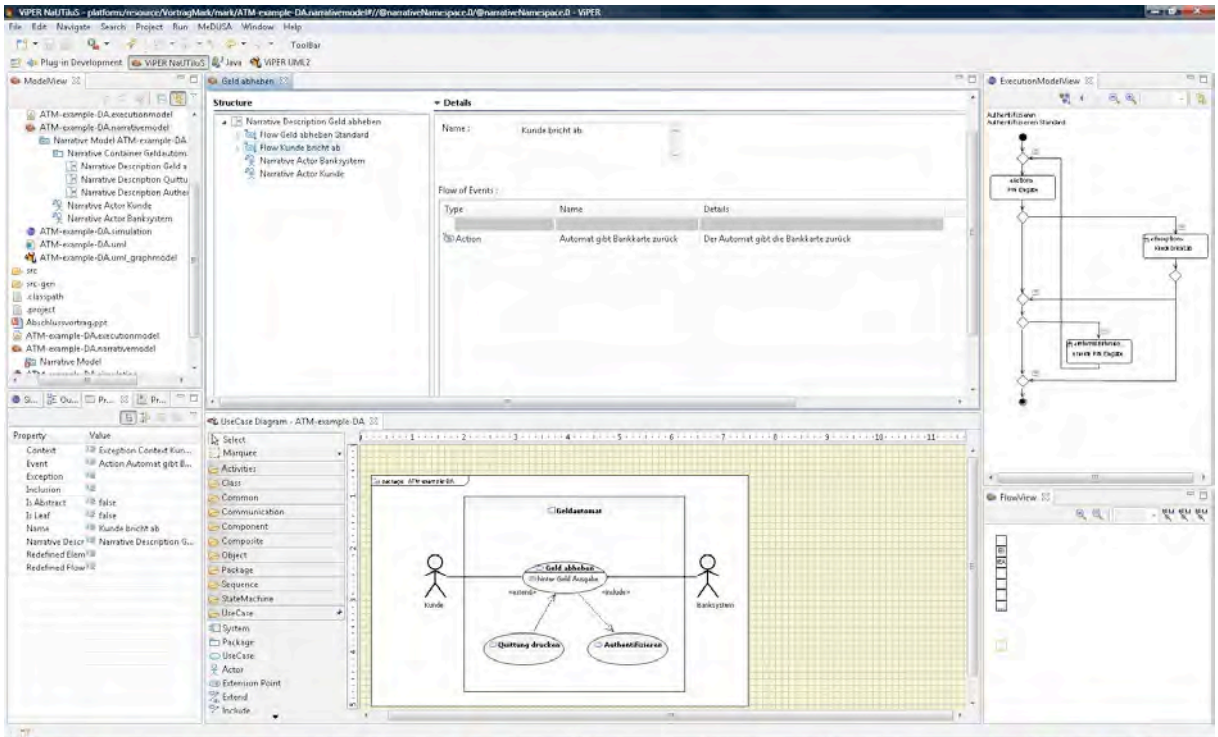
depicts the relations between the single use cases and a set of more or less detailed textual descriptions of the behaviour encapsulated by the single use cases. Although use cases have become a widespread technique, they still face several mayor acceptance issues. One of the major advantages of use case related techniques is the informal structure of the textual behaviour descriptions, which makes them easy to understand and thus a good means for communication especially between developers and customers.

This informality is at the same time their biggest drawback. Due to this lack of formality use case descriptions are difficult to maintain, since it is impossible to keep track of model completeness, as well as of consistency of the textual descriptions themselves and between textual descriptions and other requirement documents especially the use case overview diagrams automatically.

To overcome those issues our research group defined a formal meta-model for textual use case descriptions. This meta-model only affects the structure of textual behaviour descriptions, but does not restrict the language that can be used to describe the behaviour itself.

This keeps the enriched descriptions easy to understand even for readers that are unaware of the underlying meta-model, and at the same time enhances the manageability and quality of the descriptions.

Thus on the one hand the enriched descriptions remain simple enough to be usable as a communication means and on the other hand the application of the formal model enables tool support which leads more sophisticated, consistent use case descriptions. So the additional time the modeller has to spend during the specification of the behavioural descriptions is well worth the effort, especially since we think that the enriched descriptions may also be used to support the creation of other documents in the development process like test cases as well.



During the last year our research group created ViPER-NaUTiluS. ViPER-NaUTiluS adds integrated tool support for specifying, editing and analysing of enriched use case descriptions

to the ViPER-IDE. Besides simple creation and editing of use case diagrams and textual descriptions the NaUTiluS-Editor provides a lot of context information which help the developers to create behaviour specifications with a high quality.

Furthermore the NaUTiluS framework includes a use case simulator for the specified behaviour descriptions. Thus NaUTiluS enables prototyping of the system behaviour very early in the development process. Use case simulation can for example be used for reviews with domain experts or for the analysis of the different scenarios described by the Use cases.

Generating System Test Cases based on Rich Use Case Models

H.Lichter, M.Obaid

UML (Unified modelling Language) is a widely used standard of definitions and concepts to structure the use cases modelling, which forms a basic step towards well-applied software development of software systems. Although UML notational diagrams are efficient formal way to model use cases on an abstract level, natural language descriptions that capture these use cases are still necessary and widely used for requirements. The aggregation of these both UML diagrams and the detailed descriptions form the complete use case model. In the previous described research on advanced use case modelling [V.Hoffman, H.Lichter], NaUTiluS tool provides an editor for the purpose of allowing behaviour specifications of use cases with natural language descriptions. Having a high quality use case model is beneficial to represent what the future system will do and how it will behave. Such a model does not only facilitate the next development phases, but also gives the possibility to have more beneficial features. Three of more important features are System Tests, Simulation, and Architecture. This research project will focus on the generation of system test cases out of use case models.

Having high quality system test cases is very important to verify system functionality, and the integrity of software systems. Generating system test cases from requirements models is helpful to save time, effort, and money. The research focus on generating system test cases based on use case models is against two important obstacles since generation of test cases based on natural language use case descriptions is not yet possible because the underlying use case model lacks information about the steps of a use case and the relationships to other use cases. On the other hand, using a strong formal description of use cases is not possible either, because we will then ignore the important widely used natural language descriptions of use cases, and generate test cases based on an incomplete and unreadable model. NaUTiluS needs to be validated for the use within the use cases modelling if it provides enough information for our research aim (generating test cases).

In summary, we want to use or develop a use case meta-model and corresponding tool support that allow both visual and natural language descriptions of use cases in a way that it contains enough information to generate system test cases. Then, to develop a generation approach for system test cases based on such use cases model and for this approach, a suitable tool support has to be provided too.

Process Assessment based on Software Repository Data

H. Schackmann, H. Lichter

External cooperation: Kisters AG, Aachen

The development of a large portfolio of software projects raises several managerial challenges, like balancing resource allocation between different projects, and aligning development processes to the standards of the organization. Hence the project status and process quality characteristics, like planning precision or problem resolution speed, must be monitored continuously in order to identify development process weaknesses, and assess process improvements. Collecting the required data by regularly project status reporting can be expensive and intrusive, and furthermore ignores the past history of a process. This motivates mining data from routinely collected repositories like change request management (CRM) systems.

However existing CRM tools provide only a number of fixed metric evaluations and are limited in their adaptability. In order to support a flexible approach for the evaluation of metrics on CRM data, the tool *BugzillaMetrics* was developed at our group. It is based on declarative metric specifications. This enables concentrating the main effort on the model of the metric, not on its implementation.

In our ongoing work we are investigating how to apply this tool for the comparison of process quality characteristics within a project portfolio of open source projects, as well as in an industrial context.

Other Activities

- Design Patterns in Action, intense course, University Tartu, Estonia, May 21-25, *H. Lichter*
- Member of the award board “Hochschulpreis David-Kopf” 2007, *H. Lichter*
- Reviewer for dpunkt-Verlag Heidelberg and computing reviews, *H. Lichter*
- Organization of the Computer Science Department’s mentors program, *H. Lichter*
- Member of the Computer Science Department’s committee for Service-Lehre, *H. Lichter*
- Member of the examination board of Computational Material Science, *H. Lichter*
- Member of the program committee, “Modellierung 2008”, March 14-16, 2008, Berlin, *H. Lichter*
- Member of the program committee, SSIRI 2008, Second IEEE International Conference on Secure System Integration and Reliability Improvement, Yokohama, Japan, July 14-17, 2008, *H. Lichter*
- Member of the program committee, SAC 2007, The 22nd Annual ACM Symposium on Applied Computing, Track Software Engineering: Applications, Practices, and Tools, Seoul, Korea, March 11-15, 2007, *H. Lichter*
- Member of the program committee, SAC 2008, The 23nd Annual ACM Symposium on Applied Computing, Track Software Engineering: Applications, Practices, and Tools, Fortaleza, Brazil, March 16-20, 2008, *H. Lichter*
- Member of the program committee, Software Engineering 2007, Hamburg, March 27-30, 2007, *H. Lichter*
- Member of the program committee, Software Engineering 2008, Munich, February 18-22, 2007, *H. Lichter*
- Member of the program committee, “Objektorientierung, Reengineering, Architektur - Evolution komplexer Softwaresysteme”, Bremen, September 2007, *H. Lichter*
- Organization of the Beginner’s Course in Computer Science 2007, *H. Lichter, V. Hoffmann*

Teaching

In addition to undergraduate courses on Programming and Software Development the group offers on the graduate level the following set of courses focusing on Software Construction and Software Quality Assurance:

- Lecture *Software Quality Assurance and Project Management*
- Lecture *Product Line Development*
- Lecture *Object-Oriented Software Construction*
- Lecture *Managing Software Development Projects*
- Seminars and Practical Labs

Furthermore we are responsible for the Software Engineering lecture of the master program Software Systems Engineering at the Thai German Graduate School of Engineering, Bangkok, Thailand.

Talks and Publications

Talks

H. Lichter, *Bewertung von Prozessverbesserungsmaßnahmen*, KISTERS AG, Aachen, January, 2007

H. Schackmann, *An Integration Framework for Heterogeneous Automatic Software Tests*. Workshop Testing of Software - From Research to Practice, Hamburg, March 23, 2007.

H. Schackmann, *BugzillaMetrics - An adaptable tool for evaluating metric specifications on change requests.* 9th International Workshop on Principles of Software Evolution (IWPSE'07), Dubrovnik, Croatia, September 3, 2007.

A. Nyßen, *Use Case Modeling for Embedded Software Systems - Deficiencies & Workarounds.* 4th Workshop on Object-oriented Modeling of Embedded Real-Time Systems (OMER4), HNF Museums Forum, Paderborn, October 30, 2007.

H. Schackmann, *BugzillaMetrics - Design of an adaptable tool for evaluating user-defined metric specifications on change requests.* MetriKon 2007, Kaiserslautern November 15, 2007.

A. Ganser, *Test Case Structuring and Execution Control in an Integration Framework for Heterogeneous Automatic Software Tests*. Workshop Testing of Software - From Research to Practice, München, February 19, 2008.

H. Lichter, *From UML to ANSI-C An Eclipse-Based Code Generation Framework*, ICSOFT 2008, 3rd International Conference on Software and Data Technologies, Porto, Portugal, July 5-8, 2008

A. Nyßen: *MeDUSA - A Model-Based Construction Method for Embedded & Real-Time Software.* 1st IEEE International Workshop on Industrial Experience in Embedded Systems Design (IEESD 2008) Turku, Finland, July 28 - August 1, 2008.

Publications

H. Schackmann, V. Hoffmann, H. Lichter (2007): *An Integration Framework for Heterogeneous Automatic Software Tests.* Workshop Testing of Software - From Research to Practice, in Bleek, Schwentner, Züllighoven (Hrsg.) Software Engineering 2007 - Beiträge zu den Workshops, Lecture Notes in Informatics (LNI), Volume 106, GI, S. 107-112.

A. Nyßen, H. Lichter (2007): *MeDUSA - MethoD for UML2-based Design of Embedded Software Applications.* Aachener Informatik Berichte, AIB 2007-07.

T. von der Maßen (2007): *Feature-basierte Modellierung und Analyse von Variabilität in Produktlinienanforderungen*. Dissertation, RWTH Aachen. Shaker Verlag, Aachen. ISBN 978-3-8322-6027-9.

L. Grammel, H. Schackmann, H. Lichter (2007): *BugzillaMetrics - An adaptable tool for evaluating metric specifications on change requests*. Proceedings of 9th International Workshop on Principles of Software Evolution (IWPSE'07), September 3-4, 2007, Dubrovnik, Croatia, ACM Press, New York, ISBN 978-1-59593-798-8, pp 35-38.

A. Nyßen, H. Lichter (2007): *Use Case Modeling for Embedded Software Systems - Deficiencies & Workarounds.* in M. Gehrke, H. Giese, J. Stroop (Eds.) Preliminary Proceedings of the 4th Workshop on Object-oriented Modeling of Embedded Real-Time Systems (OMER4), 30.-31. October, HNF Museums Forum, Paderborn, Germany, pp 63-67.

L. Grammel, H. Schackmann, H. Lichter (2007): *BugzillaMetrics - Design of an adaptable tool for evaluating user-defined metric specifications on change requests.* MetriKon 2007 - Praxis der Software-Messung, in Büren, Bundschuh, Dumke (Hrsg.) Tagungsband des DASMA Software Metrik Kongresses MetriKon 2007, 15.-16. November 2007, Kaiserslautern, Shaker Verlag, Aachen, ISBN 978-3-8322-6703-2, pp 201-218.

A. Walter, A. Nyßen, V. Hoffmann, H. Lichter (2008): *Werkzeugunterstützung für die Use Case-Modellierung.* in K. Herrmann, B. Bruegge (Hrsg.) Software Engineering 2008, Lecture Notes in Informatics (LNI), Volume 121, GI.

A. Ganser, H. Schackmann, H. Lichter, H.-J. Schlebusch (2008): *Test Case Structuring and Execution Control in an Integration Framework for Heterogeneous Automatic Software Tests*. Workshop Testing of Software - From Research to Practice, in Software Engineering 2008 - Beiträge zu den Workshops, Lecture Notes in Informatics (LNI), Volume 122, GI, S. 387-394.

A. Nyßen, H. Lichter (2008): *The MeDUSA Reference Manual, Second Edition.* Aachener Informatik Berichte, AIB-2008-07, ISSN 0935-3232.

M. Funk, A. Nyßen, H. Lichter (2008): *FROM UML TO ANSI-C - An Eclipse-Based Code Generation Framework.* Proceedings of ICSOFT 2008, 3rd International Conference on Software and Data Technologies (ICSOFT), Porto, Portugal, Vol. SE/GSDCA/MUSE, pp 12-19, INSTICC Press, ISBN: 978-989-8111-52-4.

A. Nyßen, H. Lichter, D. Streitferdt, P. Nenninger (2008): *MeDUSA - A Model-Based Construction Method for Embedded & Real-Time Software*. 1st IEEE International Workshop on Industrial Experience in Embedded Systems Design (IEESD 2008), in Proceedings of 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC 2008), Turku, Finland.

D. Streitferdt, G. Wendt, P. Nenninger, A. Nyßen, H. Lichter (2008): *Model Driven Development Challenges in the Automation Domain.* 1st IEEE International Workshop on Industrial Experience in Embedded Systems Design (IEESD 2008), in Proceedings of 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC 2008), Turku, Finland.