REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 28-10-2007	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 1-Aug-2003 - 31-Jul-2007

4. TITLE AND SUBTITLE
Integrating Requirements Engineering, Modeling, and Verification Technologies
into Software and Systems Engineering

5a. CONTRACT NUMBER
DAAD19-03-1-0197

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Manfred Broy, Martin Leucker

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Technische Universitat Munchen
Technische Universitat Munchen
D-80290
00000 -

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
43722-CI.7

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; Federal purpose rights

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
The objective of this project is the development of an integrated suite of technologies focusing on end-to-end software development supporting requirements analysis, design, implementation, and verification. This final progress report summarizes the work that has been performed within this project. It contains an overview about the project's achievements in respect to original

15. SUBJECT TERMS
Software Engineering, Requirements Engineering, Modeling, Verification, Testing, AutoFocus, SAL, SALT, STeP

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Manfred Broy
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR		19b. TELEPHONE NUMBER +49-892-8917

Report Title

Integrating Requirements Engineering, Modeling, and Verification Technologies
into Software and Systems Engineering

ABSTRACT

The objective of this project is the development of an integrated suite of technologies focusing on end-to-end software development supporting requirements analysis, design, implementation, and verification. This final progress report summarizes the work that has been performed within this project. It contains an overview about the project's achievements in respect to original problem statement, the technical work of the related work packages, and reports on our cooperations with leading US institutes.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Bernhard Schätz, Markus Pister, and Alexander Wisspeintner.
Anforderungsanalyse in der modellbasierten Entwicklung am Beispiel von AutoFocus. Softwaretechnik-Trends, 24(1), 2004. In German.

Alexander Pretschner, Heiko Lötzbeyer, and Jan Philipps.
Model based testing in incremental system development.
In Journal of Systems and Software, volume 70, pages 315-329,
March 2004.

Number of Papers published in peer-reviewed journals: 2.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Wolfgang Prenninger and Alexander Pretschner.
Abstractions for model-based testing.
In M. Pezze, editor, Proceedings Test and Analysis of
Component-based Systems (TACoS'04), Barcelona, March 2004.

Olga Grinchtein, Bengt Jonsson, and Martin Leucker.
Learning of event-recording automata.
In Proceedings of the Joint Conferences FORMATS and FTRTFT,
volume 3253 of Lecture Notes in Computer Science, September 2004.

Therese Berg, Olga Grinchtein, Bengt Jonsson, Martin Leucker, Harald Raffelt, and Bernhard Steffen.
On the correspondence between conformance testing and regular inference.
In Maura Cerioli, editor, Fundamental Approaches to Software
Engineering, FASE'05, volume 3442 of Lecture Notes in Computer
Science, pages 175--189. Springer, 2005.

Andreas Bauer, Martin Leucker, and Jonathan Streit.
SALT-structured assertion language for temporal logic.
In Proceedings of the Eighth International Conference on Formal
Engineering Methods, Lecture Notes in Computer Science, September 2006.

Peer-Reviewed Workshops:

Leonid Kof, Bernhard Schätz, Ingomar Thaler, and Alexander Wisspeintner.
Service-based development of embedded systems.
In Net.Object Days Conference, OOSE Workshop, Erfurt, Germany,
2004.

Bernhard Schätz, Andreas Fleischmann, Eva Geisberger, and Markus Pister.
Model-based requirements engineering with autoraid.
In Proceedings of Informatik 2005 Workshop Modellbasierte
Qualitätssicherung, 2005.

Bernhard Schätz, Andreas Fleischmann, Eva Geisberger, and Markus Pister.
Modellbasierte Anforderungsentwicklung.
In Workshop "Object-Orientied Software-Engineering" (OOSE),
NetObjectDays 2005, September 2005.
In German.

Manfred Broy.
Time, abstraction, causality, and modularity in interactive systems.
In FESCA 2004. Workshop at ETAPS 2004, pages 1--8, 2004.

Olga Grinchtein, Bengt Jonsson, and Martin Leucker.
Inference of timed transition systems.
In 6th International Workshop on Verification of Infinite-State
Systems, volume 138 of Electronic Notes in Theoretical Computer
Science. Elsevier Science Publishers, 2004.

Manfred Broy, Jorge Fox, Florian Hölzl, Dagmar Koss, Marco Kuhrmann, Michael Meisinger, Birgit Penzenstadler, Sabine Rittmann,
Bernhard Schätz, Maria Spichkova, and Doris Wild.
Service-oriented modeling of cocome with focus and autofocus.
In The Common Component Modeling Example: Comparing Software
Component Models, LNCS. Springer, November 2007.
to appear.

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 10

(d) Manuscripts

Number of Manuscripts: 0.00

Number of Inventions:

Graduate Students

NAME	PERCENT_SUPPORTED
Michael Meisinger	0.25
Sabine Rittmann	0.25
Andreas Bauer	0.25
Elmar Juergens	0.25
Alexander Gruler	0.25
FTE Equivalent:	**1.25**
Total Number:	**5**

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
Dr. Eva Geisberger	0.10
Dr. Martin Leucker	0.10
Dr. Bernhard Schaetz	0.10
FTE Equivalent:	**0.30**
Total Number:	**3**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Prof. Dr. Dr. h.c. Manfred Broy	0.10	No
FTE Equivalent:	**0.10**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:......	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:......	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):......	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:......	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME	
Markus Strohmeier	
Gerrit Hanselmann	
Jonathan Streit	
Ernst Sassen	
Total Number:	**4**

Names of personnel receiving PHDs

NAME	
Leonid Kof	
Wolfgang Prenninger	
Alexander Wisspeintner	
Andreas Bauer	
Total Number:	**4**

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Integrating Requirements Engineering, Modeling and Verification Technologies into Software and Systems Engineering

Final Progress Report

Manfred Broy and
Martin Leucker

Institut für Informatik
Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany

October 28, 2007

Abstract

The objective of this project is the development of an integrated suite of technologies focusing on end-to-end software development supporting requirements analysis, design, implementation, and verification [Bro04a]. This final progress report summarizes the work that has been performed within this project. It contains an overview about the project's achievements in respect to original problem statement, the technical work of the related work packages, and reports on our cooperations with leading US institutes.

Contents

1 Problem Studied

The tight integration of requirements elicitation, validation, verification, and documentation allows iterative, or evolutionary, development processes to produce systems more efficiently than classical top-down development processes. This integration requires suitable means for organizing and interrelating the different documents that are created during development. Especially for large systems, machine support for creating and organizing these documents is desirable and necessary. The advent of powerful techniques for requirements tracing (as, e.g., exhibited by the DOORS tool) as well as recent advances in validation/verification technology let time seem ripe for an integration of these approaches into one single integrated CASE tool framework. Such a framework would allow for significantly reducing the cost of systems development.

The cost of quality assurance (active and passive) is usually recognized to be one key cost driving factor. Shorter time-to-market and more complex problems are likely to even increase the importance—and thus cost—of this factor. This is particularly true for embedded systems which are deployed in large numbers (e.g., automotive controllers or smart cards).

Commonly accepted approaches to reducing the cost of quality assurance include model based development processes, extensive testing and documentation, controlled requirements tracing, and the application of sophisticated verification and test case generation techniques.

The projects aims at fostering *cooperations* with renowned research institutes in the US, in particular with the Stanford Research Institute and Prof. Zohar Manna´s REACT Group at Stanford University, in the field of embedded systems design based on synchronous, time-triggered architectures. The goal is to develop an *integrated approach* to modeling, tracing, and verifying embedded automotive systems, together with dedicated *tool support*. It is expected that the prospective results carry over to other application domains as well.

The project main topics include: *Requirements tracing*, *verification techniques*, and *testing* for synchronous systems.

Requirements Tracing An entire process for requirements tracing needs an integration of a modeling tool (e. g. AutoFOCUS) and a requirements management tool (e. g. *DOORS* from Telelogic, one of the most popular requirements management tools available). Such an integrated toolbox would be able to be used for capturing and structuring requirements. It would provide support for incremental system development. The relations between informal requirements, formal requirements, test cases, verification properties and system design would be documented and it is possible to validate whether the requirements are fulfilled by the system design or not. This advanced tool support would lead to less errors in system analysis, design and implementation resulting in lower system development costs.

Verification of Synchronous Systems To support the verification approach of quality assurance in the development process, an integration of the modeling tool (AutoFOCUS) and the requirements management tool (DOORS) together with verification tools (automatic and interactive) is necessary.

An integrated tool box must support the export of models into formal theories of the verification tool and the translation of suitable classes of requirements (universal properties) from the requirements management tool into temporal logic and predicate logic specifications.

In addition to the implementation work, preliminary theoretical work must determine formalizations of synchronous models that are amenable to interactive verification. In addition, abstraction techniques must be explored to make verification feasible.

Testing Synchronous Systems Because of gaps in a formally based development process, such as interactions of software with hardware devices or the integration of a system into an existing one, mathematically established propositions on the level of models yet have to be complemented by test cases for the actual implementation. In addition, the complexity of industrial systems makes them not always amenable to a complete mathematical analysis (e.g., the state space explosion problem). Specialized techniques for deriving test sequences, together with a methodologically founded strategy for selecting test sequences, is thus a necessary supplement to verification technologies.

2 Summary of Most Important Results

In this section, we highlight the most important results obtained in each of the research areas *Requirements tracing*, *verification techniques*, and *testing* for synchronous systems. Furthermore, we report on case studies carried out, summarize the project's documentation, and describe the cooperation with leading US research institutes and universities established and strengthened within the project.

2.1 Requirements Tracing

We have defined a requirements engineering process that delivers and maintains a formal specification, realized tools supporting this process, and worked out modeling formalisms for requirements.

Integrated Development Process We worked out the definition of a requirements engineering process that delivers a formal specification and allows maintenance of the specification. The task of requirements engineering is to find a way from the informal and unstructured requirements to a precise (formal) and structured description of the system to be developed. We developed a set of concepts and developed a tool to support a process using these concepts. This tool is strongly integrated into the AutoFocus tool. The process has following iterative steps (need not to be performed in this order):

- *Getting Requirements.* The requirements are elicited. This can be done by interviews, workshops etc.

- *Refinement and Development.* The given information is structured in a refinement relation. The task is here to justify the existence of requirements. The sources of these relations are business goals.
- *Structuring Requirements.* The requirements are structured by classification and by model elements. According to their classification the requirements can be connected to formal model elements. They can either motivate the existence of an element or they can describe a property of the element.
- *Analysis and Completion/Correction.* The models are used to guide the requirements engineers to ask questions regarding the completeness and the consistency of the specification and thus revise the specification.

The method has been developed for the following views: *Structure*, *model*, and *data-type*.

We have developed the specification framework FIRE *("Formal and Informal Requirements Embedding")* that integrates informal and formal requirements, defines their relationships, and sets the foundation for requirements pre- and post-tracing. Within that framework, we have focused on the details of formally capturing requirements; therefore we have developed a formal language, which bases on the AutoFOCUS formalism [HSSS96, SH99], and which extends it with new elements and mechanisms. The interaction view of AutoFOCUS (sequence charts) was extended and adapted to the project's scope. This allows formalizing many types of functional requirements.

The requirements engineering process FIRE and the requirements tracing tool support AutoRAID [Tea04b] was documented by publications:

- Bernhard Schätz, Markus Pister, and Alexander Wisspeintner. Anforderungsanalyse in der modellbasierten Entwicklung am Beispiel von AutoFocus. *Softwaretechnik-Trends*, 24(1), 2004. In German
- Bernhard Schätz, Andreas Fleischmann, Eva Geisberger, and Markus Pister. Model-based requirements engineering with autoraid. In *Proceedings of Informatik 2005 Workshop Modellbasierte Qualitätssicherung*, 2005
- Bernhard Schätz, Andreas Fleischmann, Eva Geisberger, and Markus Pister. Modellbasierte Anforderungsentwicklung. In *Workshop*

"Object-Orientied Software-Engineering" (OOSE), NetObjectDays 2005, September 2005. In German

Requirements Tracing Tool Integration Prototype In the project proposal we have suggested to realize an integration of the requirements management tool DOORS from Telelogic and the CASE tool AutoFocus to allow for requirements tracing between textual requirements and AutoFocus model elements. After investigating the technical possibilities for integrating the tools we concluded that a tight integration using one joint data repository is not possible.

In consequence of this fact, the proposed requirements engineering functionality was realized directly as part of our AutoFocus CASE tool. The AutoRaid extension [Tea04b] of AutoFocus has been built. For a documentation of AutoRaid, see [Tea04a].

Modeling Formal Requirements We developed a new diagram type – the *Service Configuration Diagram* – that is integrated into our description technique AutoFocus. This diagram type is used to describe changes of active functionalities during system execution. In [KSTW04] we have described a service based modeling process using the *Service Configuration Diagrams*. In contrast to most component based approaches our method focuses on identifying single system functionalities. The modeling process is applicable during both requirements analysis and design phase. The modeling process together with the AutoFocus description technique and the new Service Configuration Diagrams are suitable for modeling user requirements in a precise way.

We further worked on semi-automatically extracting formal ontologies from informal requirement specifications. The work is presented in the PhD. thesis of Leonid Kof:

- Leonid Kof. *Text Analysis for Requirements Engineering*. PhD thesis, Technische Universität München, 2005

The AutoFocus semantics is message asynchronous and time synchronous according to the classification given in [SBHW03]. The time synchrony is responsible for the existence of implicit time constraints in the AutoFocus models. The time constraints are reasonable within design models but within the context of requirements analysis models, it is often desirable to abstract from time and only consider the core functionalities.

We decided to develop a time asynchronous semantics for the classic AutoFOCUS diagram types *System Structure Diagram (SSD)* and *State Transition Diagram (STD)*. This semantics is based on the given time synchronous AutoFOCUS semantics, abstracts from time constraints and is suitable for modeling requirements. The semantics is formalized using a translation between AutoFOCUS and the specification language Focus [BS01].

The core ideas of the new semantics were published in:

- Manfred Broy. Time, abstraction, causality, and modularity in interactive systems. In *FESCA 2004. Workshop at ETAPS 2004*, pages 1–8, 2004

The formal specification language SALT was developed and documented mainly within the master thesis by Jonathan Streit [Str06]:

- Jonathan Streit. Development of a programming language like temporal logic specification language. Master's thesis, Fakultät für Informatik, Technische Universität München, 2006. URL `http://salt.in.tum.de`

Its gist was presented to the research community in the following publication [BLS06]:

- Andreas Bauer, Martin Leucker, and Jonathan Streit. SALT—structured assertion language for temporal logic. In *Proceedings of the Eighth International Conference on Formal Engineering Methods*, Lecture Notes in Computer Science, September 2006

The work and its extensions is extensively documented on SALT's homepage at `salt.in.tum.de`.

The time asynchronous semantics for the classic AutoFOCUS diagram types *System Structure Diagram (SSD)* and *State Transition Diagram (STD)* were formalized and documented in the dissertation (Ph.D. thesis) of Alexander Wißpeintner [Wiß06]:

- Alexander Wißpeintner. *Verhaltensinvariante Transformation von Entwurfsmodellen Reaktiver Systeme – Eine Adaption der Refactoring-Technik auf gezeitete Modelle unter Verwendung eines formalen Verhaltensäquivalenzbegriffs*. Dissertation, Fakultät für Informatik, Technische Universität München, 2006. In German

2.2 Verification of Synchronous Systems

For verifying synchronous systems we anticipated the integration of two established verification tools in the AutoFOCUS framework, namely PVS (from SRI International, [Rus97, OS99]) and STeP (from Stanford University, [MtSg95]). After consulting SRI International we decided to realize an AutoFOCUS-SAL translation instead of an AutoFOCUS-PVS translation. The reasons for this decision are given in the following paragraphs. This translation was designed during a one month stay at SRI International in Menlo Park (see Section 2.6) in close collaboration with the respective provider of the verification tool.

SAL [Sha00, BGL^{+}00, dMOS03] provides a very powerful verification environment for synchronous and asynchronous systems by combining model checking with decision procedures (ICS, [FORS01]). Therefore SAL is particularly suitable for performing verification tasks on AutoFOCUS-models and we expect significantly better results from an AutoFOCUS-SAL-STeP integration than from an AutoFOCUS-PVS-STeP integration. However, an AutoFOCUS-PVS translation might be a future issue and is presumably realized by SRI International as a PVS-SAL integration [For03].

The AutoFOCUS-SAL integration is straightforward, as SAL supports synchronous composition and the required data type constructs. The design of the translation from AutoFOCUS to SAL was constructed in collaboration with the Computer Science Laboratory of SRI International. It is documented in [Wis06] and we have prototypically implemented the translation within the AutoFOCUS-Quest framework.

Concerning the anticipated STeP [MtSg95] translation, the synchronous AutoFOCUS models have to be transformed into asynchronously communicating STeP models with an appropriate synchronization mechanism. For the actual translation we choose an interleaving model for representing parallel composed AutoFOCUS components, as an explicit generation of the cross product of the component's transition relation leads to unsuitable large transitions. Due to the use of the interleaving composition, the verification properties have to be strengthened in order to hold for the interleaving system. In collaboration with Zohar Manna's group from Stanford University, we performed experiments toward systematically strengthening properties for inductive proofs on interleaving system models.

In general, however, interactive verification approaches turn out to be time consuming. We therefore decided to extend AutoFocus's automatic and lightweight verification capabilities. Within the work topic "verification tools" we integrated the explicit state model checker *SPIN* [Hol97, Hol03] into the AutoFocus CASE tool. The work was partly done within the following master thesis:

- Markus Strohmeier. Modellbasierte Validierung verteilter Komponenten: Kopplung von AutoFocus und SPIN. Diplomarbeit, Technische Universität München, January 2005. In German

Furthermore, we integrated Stanford's *LOLA* system [DSS$^+$05] for *runtime verification* into AutoFocus, allowing to find bugs while simulating models, even when model checking approaches fail due to too large state spaces. Additionally, Andreas Bauer examined extensions to runtime verification, especially in his thesis:

- Andreas Bauer. *Model-based runtime analysis of distributed reactive systems*. PhD thesis, Institut für Informatik, Technische Universität München, 2007

Abstraction techniques in the context of AutoFocus have been worked out in the following master thesis [Sas06]:

- Ernst Sassen. Abstrakte Modellinterpretation: Design und prototypische Implementierung eines Abstrakten Modell-Interpreters. Master's thesis, Fakultät für Informatik, Technische Universität München, 2006

2.3 Testing Synchronous Systems

In [PLP04] we presented a CLP (Constraint Logic Programming) based test case generator integrated into the AutoFocus CASE tool.

The CLP based test case generator was used to extract test cases from a formal communication protocol specification. These test cases were used to validate the informal parts of the protocol specification with the aim to identify ambiguities. Furthermore the test cases were used to carry out hardware in the loop tests of automotive control units.

In [PP04] we classified and discussed different kinds of abstraction for building test models and using these models for test case generation.

The CLP based test case generator was extended with a strategy for storing sets of states to enhance the efficiency of the initial version. This technology avoids certain loops in the search algorithm and therefore prevents the test case generator from running into infinite loops.

A process for developing special test models based on the AutoFOCUS modeling language was established. The test models are used to automatically derive test cases by applying the AutoFOCUS test case generator. Different abstraction techniques are used to derive test models from existing specification models.

The results of the work are documented in the following dissertation (PhD. thesis):

- Wolfgang Ludwig Johann Prenninger. *Inkrementelle Entwicklung von Verhaltensmodellen zum Test von reaktiven Systemen*. Dissertation, Technische Universität München, July 2005. In German

As an alternative to the test case generator developed by the AutoFOCUS team, the SAL test case generator [HdMR04, HMR05] has also been integrated into AutoFOCUS. For this, translations from AutoFOCUS models into SAL specifications are enriched by trap variables denoting the test goal. Furthermore, SAL test cases have to be translated back to AutoFOCUS' data structures.

The concept of this approach is documented in [Wis06]:

- Alexander Wisspeintner. Using the SAL automated test case generator on AutoFocus models. Technical note, Fakultät für Informatik, Technische Universität München, May 2006

The migration between existential and universal properties has been addressed using automata learning techniques. Such learning techniques allow to derive models comprising the complete system behavior, if only some system behavior is given. Learning techniques for timed systems have been developed in [GJL04b, GJL04a]:

- Olga Grinchtein, Bengt Jonsson, and Martin Leucker. Learning of event-recording automata. In *Proceedings of the Joint Conferences FORMATS and FTRTFT*, volume 3253 of *Lecture Notes in Computer Science*, September 2004
- Olga Grinchtein, Bengt Jonsson, and Martin Leucker. Inference of timed transition systems. In *6th International Workshop on Verification of Infinite-State Systems*, volume 138 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2004

Interestingly, conformance test suites and exemplifying system behavior for learning may coincide, as shown in [BGJ$^+$05]:

- Therese Berg, Olga Grinchtein, Bengt Jonsson, Martin Leucker, Harald Raffelt, and Bernhard Steffen. On the correspondence between conformance testing and regular inference. In Maura Cerioli, editor, *Fundamental Approaches to Software Engineering, FASE'05*, volume 3442 of *Lecture Notes in Computer Science*, pages 175–189. Springer, 2005

2.4 Case Study

We have used a small traffic lights specification as running example for illustrating the design of the AF-STeP and AF-SAL integrations. As real drive-by-wire applications have not been realized in practice yet, we based the evaluation of the requirements of the requirements engineering process on a specification of a car control unit of DaimlerChrysler [HP02]. The specification describes a door control system realizing several comfort functions, for example central locking, electronic window lift and electronic seat adjustment.

The service based modeling process was applied to model the electronic seat adjustment system of a car. The case study is documented in [KSTW04].

The test process described in the previous section was applied in a concrete case study. Subject of the case study was testing the network master device of a MOST (media oriented systems transport) network, a new network standard for automotive multimedia applications. The results of the case study are documented in [Pre05, Chapter 7].

We formulated a security state model of an EMV CPA Card Application [EMV05] in AutoFOCUS for deriving test cases showing conformance of an implementation with the standard. For this, we applied the AutoFOCUS-SAL test case generator.

In the Common Component Modeling Example Contest [BFH+07] we showed how to specify and develop a cash desk application, which is a typical distributed system consisting of embedded controllers (e. g. a credit card reader or the barcode scanner) as well as components for data storage (e. g. the inventory). For specification and development we were using AutoFOCUS with its component-oriented FOCUS based approach. There we applied a rigorous development process based on different levels of abstraction, which trace from requirements to implementation. These comprise the partial behavior descriptions of application services, total behavior descriptions of logical components and the deployment of the complete system in a defined execution environment, which resembles the FOCUS semantics. The implementation was gathered by code generation from the AutoFOCUS model. The results of this case study are described in:

- Manfred Broy, Jorge Fox, Florian Hölzl, Dagmar Koss, Marco Kuhrmann, Michael Meisinger, Birgit Penzenstadler, Sabine Rittmann, Bernhard Schätz, Maria Spichkova, and Doris Wild. Service-oriented modeling of cocome with focus and autofocus. In *The Common Component Modeling Example: Comparing Software Component Models*, LNCS. Springer, November 2007. to appear

2.5 Project Documentation

The project documentation evolved with the project progress. In total

- 2 journal papers [SPW04, PLP04],
- 4 conference papers [PP04, GJL04b, BGJ+05, BLS06],
- 6 workshop papers [KSTW04, SFGP05a, SFGP05b, Bro04b, GJL04a, BFH+07],
- 1 technical report [Wis06],
- 4 PhD theses [Kof05, Pre05, Wiß06, Bau07],

- 4 MSc theses [Str05, Han05, Str06, Sas06], and
- 1 BSc thesis [Fab05]

have been written by our group with (partial) support by the project.

The AutoRAID tool is further described in [Tea04a, Tea04b]. SALT is documented also on SALT's homepage at `salt.in.tum.de`.

2.6 Cooperation with US Institutes

A side objective of the project was the establishment of cooperations with leading US institutes in the project's research areas. Within a one month stay of the two research associates *Heiko Loetzbeyer* and *Alexander Wisspeintner* at SRI International in Menlo Park, we could initiate tighter cooperations with SRI International and Stanford University. *Martin Leucker* was visiting SRI too, as well as Stanford University (REACT Group). Furthermore *Bernhard Schätz* was visiting the Center for Hybrid and Embedded Software Systems at the University of California, Berkeley.

SRI International:

In collaboration with the Computer Science Laboratory of SRI International we designed the AutoFOCUS-SAL translation and examined the use of SAL technology for test case generation. Thanks to *Patrick Lincoln*, *John Rushby*, *Natarajan Shankar*, *Leonardo de Moura*, *Ashish Tiwari* and *Gregoire Hamon* for their contribution to the translation and the many discussions we had during our visit at SRI International. While working at SRI, we experienced a strong commitment of SRI to our common goals and intend to continue our cooperation in future.

Stanford University (REACT Group):

The design of AutoFOCUS-STeP translation stems from a cooperation with *Zohar Manna's* REACT Group at Stanford University. *Henny Sipma* and *Matteo Slanina* made an excellent contribution to the translation of AutoFOCUS models and properties to STeP. Furthermore we investigated proof tactics in STeP to verify AutoFocus models.

Moreover, we learned in detail about the REACT Group's activities in runtime verification, resulting in the tool *LOLA*. We found out that LOLA actually fits nicely to complement AutoFOCUS' heavy-weight verification techniques by so-called light-weight verification techniques. Special thanks goes to Zohar Manna, Henny Sipma, and Cesar Sanchez for their support.

Center for Hybrid and Embedded Software Systems (University of California at Berkeley):

We also visited *Edward Lee* and his group in Berkeley. Edward Lee's group has prime expertise in embedded software development with special emphasis on visualization and simulation of hybrid systems. Especially within an one month stay of our team member Bernhard Schätz in Berkeley it was possible to compare the different semantics of the *Ptolemy II* CASE tool and the AutoFocus CASE tool. We realized a translation of AutoFocus models into Ptolemy-II models. This work was partly done within the following bachelor thesis:

- Stephan Fabrizek. Evaluierung und Realisierung eines Übergangs von der AutoFocus Semantik in das Ptolemy-Framework. Bachelor thesis, Technische Universität München, 2005. In German

Moreover, Martin Leucker visited Koushik Shen at Berkeley, who also works on runtime verification topics.

Department of Computer Science,University of California, Santa Cruz:

Martin Leucker visited *Luca de Alfaro*, who works (among other things) on abstraction techniques in the context of formal verification.

Department of Computer Science and Engineering, University of California, San Diego:

Ingolf Krüger from the University of California, San Diego visited our group in Munich several times. We are cooperating in defining a service based approach for software system development.

Siemens Corporate Research, Princeton:

We cooperated with Siemens Corporate Research at Princeton. *Gerrit Hanselmann* worked at the site at Princeton on automated test case generation and test execution using the UML testing profile. He received the master degree for his work:

- Gerrit Hanselmann. An approach for generating and executing tests based on the uml testing profile. Diplomarbeit (master thesis), Technische Univeristät München, 2005

References

[Bau07] Andreas Bauer. *Model-based runtime analysis of distributed reactive systems*. PhD thesis, Institut für Informatik, Technische Universität München, 2007.

[BFH$^+$07] Manfred Broy, Jorge Fox, Florian Hölzl, Dagmar Koss, Marco Kuhrmann, Michael Meisinger, Birgit Penzenstadler, Sabine Rittmann, Bernhard Schätz, Maria Spichkova, and Doris Wild. Service-oriented modeling of cocome with focus and autofocus. In *The Common Component Modeling Example: Comparing Software Component Models*, LNCS. Springer, November 2007. to appear.

[BGJ$^+$05] Therese Berg, Olga Grinchtein, Bengt Jonsson, Martin Leucker, Harald Raffelt, and Bernhard Steffen. On the correspondence between conformance testing and regular inference. In Maura Cerioli, editor, *Fundamental Approaches to Software Engineering, FASE'05*, volume 3442 of *Lecture Notes in Computer Science*, pages 175–189. Springer, 2005.

[BGL$^+$00] Saddek Bensalem, Vijay Ganesh, Yassine Lakhnech, César Muñoz, Sam Owre, Harald Rueß, John Rushby, Vlad Rusu, Hassen Saïdi, N. Shankar, Eli Singerman, and Ashish Tiwari. An overview of SAL. In C. Michael Holloway, editor, *LFM 2000: Fifth NASA Langley Formal Methods Workshop*, pages 187–196, Hampton, VA, jun 2000. NASA Langley Research Center.

[BLS06] Andreas Bauer, Martin Leucker, and Jonathan Streit. SALT—structured assertion language for temporal logic. In *Proceedings of the Eighth International Conference on Formal Engineering Methods*, Lecture Notes in Computer Science, September 2006.

[Bro04a] Manfred Broy. Architecture driven modeling in software development. In *Proceedings of the Nineth International Conference on Engineering of Complex Computer Systems*, pages 3–14. IEEE Computer Society, 2004.

[Bro04b] Manfred Broy. Time, abstraction, causality, and modularity in interactive systems. In *FESCA 2004. Workshop at ETAPS 2004*, pages 1–8, 2004.

[BS01] Manfred Broy and Ketil Stølen. *Specification and Development of Interactive Systems – Focus on Streams, Interfaces, and Refinement*. Monographs in Computer Science. Springer-Verlag New York, Inc., 2001.

[dMOS03] Leonardo de Moura, Sam Owre, and Natarajan Shankar. The SAL language manual. Technical Report SRI-CSL-01-02, Computer Science Laboratory, SRI International, Menlo Park, CA, 2003.

[DSS$^+$05] Ben D'Angelo, Sriram Sankaranarayanan, Cesar Sanchez, Will Robinson, Bernd Finkbeiner, Henny B. Sipma, Sandeep Mehrotra, and Zohar Manna. LOLA: Runtime monitoring of synchronous systems. In *TIME '05: Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)*, pages 166–174, Washington, DC, USA, 2005. IEEE Computer Society.

[EMV05] EMVCo. Emv common payment application 1.0, December 2005. URL `http://www.emvco.com/cgi_bin/terms.pl?action=down&fichier=documents/specification/download/EMVCommonPaymentApplicationSpecificationv1Dec2005_PU.zip`.

[Fab05] Stephan Fabrizek. Evaluierung und Realisierung eines Übergangs von der AutoFocus Semantik in das Ptolemy-Framework. Bachelor thesis, Technische Universität München, 2005. In German.

[For03] Formal Methods Program. Formal methods roadmap: PVS, ICS, and SAL. Technical Report SRI-CSL-03-05, Computer Science Laboratory, SRI International, Menlo Park, CA, October 2003.

[FORS01] Jean-Christophe Filliâtre, Sam Owre, Harald Rueß, and Natarajan Shankar. ICS: Integrated canonizer and solver. In *CAV '01: Proceedings of the 13th International Conference on Computer Aided Verification*, pages 246–249, London, UK, 2001. Springer-Verlag.

[GJL04a] Olga Grinchtein, Bengt Jonsson, and Martin Leucker. Inference of timed transition systems. In *6th International Workshop on Verification of Infinite-State Systems*, volume 138 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2004.

[GJL04b] Olga Grinchtein, Bengt Jonsson, and Martin Leucker. Learning of event-recording automata. In *Proceedings of the Joint Conferences FORMATS and FTRTFT*, volume 3253 of *Lecture Notes in Computer Science*, September 2004.

[Han05] Gerrit Hanselmann. An approach for generating and executing tests based on the uml testing profile. Diplomarbeit (master thesis), Technische Univeristät München, 2005.

[HdMR04] Grégoire Hamon, Leonardo Mendonça de Moura, and John M. Rushby. Generating efficient test sets with a model checker. In *SEFM*, pages 261–270. IEEE Computer Society, 2004.

[HMR05] Grégoire Hamon, Leonardo De Moura, and John Rushby. Automated test generation with SAL. Csl technical note, SRI International, January 2005. URL `http://www.csl.sri.com/users/rushby/papers/testgen.ps.gz`.

[Hol97] Gerard J. Holzmann. The model checker SPIN. *Software Engineering*, 23(5):279–295, 1997.

[Hol03] Gerard J. Holzmann. *The SPIN Model Checker*. Pearson Education, 2003.

[HP02] Frank Houdek and Barbara Paech. Das Türsteuergerät - eine Beispielspezifikation. Technical Report IESE-Report Nr. 002.02/D, Fraunhofer Institut Experimentelles Software Engineering (IESE), 2002.

[HSSS96] Franz Huber, Bernhard Schätz, Alexander Schmidt, and Katharina Spies. Autofocus - a tool for distributed systems specification. In Bengt Jonsson and Joachim Parrow, editors, *FTRTFT'96 - Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 1135 of *Lecture Notes in Computer Science*, pages 467–470. Springer Verlag, 1996.

[Kof05] Leonid Kof. *Text Analysis for Requirements Engineering*. PhD thesis, Technische Universität München, 2005.

[KSTW04] Leonid Kof, Bernhard Schätz, Ingomar Thaler, and Alexander Wisspeintner. Service-based development of embedded systems. In *Net.Object Days Conference, OOSE Workshop, Erfurt, Germany*, 2004. Electronic Publication, `http://www.netobjectdays.org/node04/de/Conf/publish/talks.html`.

[MtSg95] Zohar Manna and the STeP group. Step: The stanford temporal prover (educational release), user's manual. Technical Report STAN-CS-TR-95-1562, November 1995.

[OS99] Sam Owre and Natarajan Shankar. The formal semantics of PVS. March 1999.

[PLP04] Alexander Pretschner, Heiko Lötzbeyer, and Jan Philipps. Model based testing in incremental system development. In *Journal of Systems and Software*, volume 70, pages 315–329, March 2004.

[PP04] Wolfgang Prenninger and Alexander Pretschner. Abstractions for model-based testing. In M. Pezze, editor, *Proceedings Test and Analysis of Component-based Systems (TACoS'04)*, Barcelona, March 2004.

[Pre05] Wolfgang Ludwig Johann Prenninger. *Inkrementelle Entwicklung von Verhaltensmodellen zum Test von reaktiven Systemen*. Dissertation, Technische Universität München, July 2005. In German.

[Rus97] John Rushby. Specification, proof checking, and model checking for protocols and distributed systems with PVS. Tutorial presented at {FORTE X/PSTV XVII '97}, November 1997.

[Sas06] Ernst Sassen. Abstrakte Modellinterpretation: Design und prototypische Implementierung eines Abstrakten Modell-Interpreters. Master's thesis, Fakultät für Informatik, Technische Universität München, 2006.

[SBHW03] Bernhard Schätz, Peter Braun, Franz Huber, and Alexander Wisspeintner. Consistency in model-based development. In *Proceedings of ECBS 2003 10th IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, 2003.

[SFGP05a] Bernhard Schätz, Andreas Fleischmann, Eva Geisberger, and Markus Pister. Model-based requirements engineering with autoraid. In *Proceedings of Informatik 2005 Workshop Modellbasierte Qualitätssicherung*, 2005.

[SFGP05b] Bernhard Schätz, Andreas Fleischmann, Eva Geisberger, and Markus Pister. Modellbasierte Anforderungsentwicklung. In *Workshop "Object-Orientied Software-Engineering" (OOSE), NetObjectDays 2005*, September 2005. In German.

[SH99] Bernhard Schätz and Franz Huber. Integrating formal description techniques. In Jeannette M. Wing, Jim Woodcock, and Jim Davies, editors, *FM'99 – Formal Methods, Proceedings of the World Congress on Formal Methods in the Development of Computing Systems, Volume II*, volume 1709 of *Lecture Notes in Computer Science*, pages 1206–1225. Springer Verlag, September 1999.

[Sha00] Natarajan Shankar. Combining theorem proving and model checking through symbolic analysis. In *CONCUR'00: Concurrency Theory*, number 1877 in Lecture Notes in Computer Science, pages 1–16, State College, PA, aug 2000. Springer-Verlag.

[SPW04] Bernhard Schätz, Markus Pister, and Alexander Wisspeintner. Anforderungsanalyse in der modellbasierten Entwicklung am Beispiel von AutoFocus. *Softwaretechnik-Trends*, 24(1), 2004. In German.

[Str05] Markus Strohmeier. Modellbasierte Validierung verteilter Komponenten: Kopplung von AutoFocus und SPIN. Diplomarbeit, Technische Universität München, January 2005. In German.

[Str06] Jonathan Streit. Development of a programming language like temporal logic specification language. Master's thesis, Fakultät für Informatik, Technische Universität München, 2006. URL `http://salt.in.tum.de`.

[Tea04a] AutoRAID Team. The AutoRAID Documentation, 2004. `http://www4.in.tum.de/~autoraid/Dokumentation/autoraid-documentation.zip`.

[Tea04b] AutoRAID Team. The AutoRAID Web Page, 2004. `http://www4.in.tum.de/~autoraid`.

[Wis06] Alexander Wisspeintner. Using the SAL automated test case generator on AutoFocus models. Technical note, Fakultät für Informatik, Technische Universität München, May 2006.

[Wiß06] Alexander Wißpeintner. *Verhaltensinvariante Transformation von Entwurfsmodellen Reaktiver Systeme – Eine Adaption der Refactoring-Technik auf gezeitete Modelle unter Verwendung eines formalen Verhaltensäquivalenzbegriffs*. Dissertation, Fakultät für Informatik, Technische Universität München, 2006. In German.