Bibliography


Bibliography


Bibliography


[34] Steve Cook, Anneke Kleppe, Richard Mitchell, Bernhard Rumpe, Jos B. Warmer, and Alan Wills. The Amsterdam manifesto on OCL. In Clark and Warmer [26], pages 115–149.


Bibliography


Bibliography


186


Bibliography


Bibliography


Bibliography


Bibliography


Summary

Embedded real-time systems are small computer systems which are used to control an increasing number of devices in every-day life. They are embedded in, for example, DVD players, microwave ovens, antilock braking systems, and autopilots. It is important that these devices always perform their function correctly in case the life of people depends upon the software used in them. Moreover, high costs are usually involved in recalling defective devices, for example, in cars. Therefore, it is desirable that these systems are formally validated, that is, a proof of the correct functioning of the system is constructed. Such a proof is especially important for real-time systems, because they not only need to function correctly, but also deliver their reactions on time. For example, an air-bag should not only inflate when a car crashes, but it should inflate milliseconds after the impact, and not seconds.

Ever since the first embedded systems were developed, their complexity has been steadily increasing. In order to control and understand this complexity different methods are used to describe the structure, the behaviour, and the requirements of software systems. Such methods are provided by the Unified Modelling Language (UML) and its Object Constraint Language (OCL) as notations (as diagrams) for describing complex object-oriented software systems, where the parts of these systems during execution are called objects. Objects react to messages they exchange among each other and with their environment, that is, with their external world. This exchange of messages is considered to be (part of) their behaviour.

UML provides the schema language of class diagrams for describing the structure, that is, the parts of the system and which parts may communicate with each other, and the notation of state machines for describing the behaviour of a system or its parts. OCL is used to describes the requirements on the system. Requirements are the properties a system has to satisfy and describe its correct functioning from the point of view of these given requirements.

In order to enable the development and the formal validation of these systems we have to define a formal semantics for the notations of UML and OCL. This means, we assign a precise meaning to the constructs of UML and OCL. This is necessary, because at present UML notations have no precise meaning. To this end, we define an unambiguous subset of UML class diagrams and define a precise mathematical semantics for this subset in Chapter 2.

UML and OCL are typed languages. This means that there are so-called typing rules on diagrams and expressions which describe when they are well-formed and therefore
Summary

have a meaning that makes sense. In Chapter 3 we show that these rules are too in-
flexible for writing requirements while the system is still under development. Namely,
development causes changes in the system which, according to the typing rules, un-
extectedly render requirements ill-formed. As a consequence, these requirements are
considered nonsensical in UML. However, in our semantics they have a well-defined
meaning, which has not been changed by the development step. To overcome this prob-
lem we propose extensions of the typing rules (based on so-called intersection types,
union types, and bounded operator abstraction) which also improve the integration of
the OCL into the UML, and which considers more requirements as well-formed.

We use logic to formalise the meaning of UML diagrams and OCL expressions
in order to enable their formal validation. Logic makes the use of interactive theorem
provers possible. Theorem provers assist in constructing proofs of the correct function-
ing of systems. This means that a system and its requirements have to be translated
into logic. The result of this translation should be of a form that allows one to exploit
all automated reasoning facilities offered by the theorem prover in finding a proof, be-
cause otherwise the construction of proofs quickly becomes complex, burdensome, and
(economically) infeasible. In Chapter 4 we describe such a translation, performed by a
computer program, into the input language of the theorem prover PVS and show why
the translator preserves the meaning of the system and its requirements.

In order to support the specification of systems during early stages of design, we
have analysed the semantics of OCL Message Expressions in Chapter 5. Message ex-
pressions specify whether messages have been sent by objects. These have been found
to be inadequate. Therefore, we propose introducing history variables to OCL. His-
tory variables allow not only to specify and reason about the messages sent during
the invocation of an operation, but also about the history of all messages sent and re-
ceived by an object. We also show that everything which can be expressed by message
expressions can also be expressed with history variables.

We strictly separate local specifications, which are requirements on the internal state
of objects (and play the role of so-called data invariants), from local behavioural spec-
fications, which describe the messages sent and received by an object. At a third level,
we introduce global specifications which specify how objects in a system may interact.

This formalisation leads to a compositional history-based specification formalism,
for which we give a compositional proof rule in Chapter 6. A specification is called
compositional if the function of a system can be derived from the functions of its parts
and the way they are put together. The main problem to solve here is the treatment
of the evolution of object structures. Object structures change because objects learn
about other objects during their lifetime, which enables them to communicate with
new acquaintances; especially, when objects create new objects.

Finally, in Chapter 7 we extend this history-based formalism to real-time specifi-
cations. We specify a part of a medium altitude reconnaissance system, which is deployed
by the Royal Dutch Air-Force, and prove its correctness. This example shows that the
methods described in this thesis can be applied in principle to real-world case studies.
Samenvatting

Ingebedde real-time systemen zijn (kleine) computer systemen die ertoe dienen de apparaten waarin ze ingebed zijn te helpen (be)sturen. Voorbeelden van zulke apparaten zijn DVD spelers, automatische remmen, autopiloten, mobiele telefoons en Magnetic-Resonance scanners. Zulke ingebedde systemen komen meer en meer voor en worden in hoog tempo snel complexer. Ook komt het steeds vaker voor dat mensenlevens van het correct functioneren van de door hen gestuurde apparaten afhangen. Deze ontwikkeling is niet meer te stuiten. Daarom is het belangrijk dat zulke apparaten correct functioneren. En dat hangt weer af van het correcte functioneren van de hen sturende real-time systemen.

Aangezien deze systemen alom tegenwoordig zijn, zijn er industriële standaards ontwikkeld om hun functionaliteit te beschrijven. Een veel gebruikte standaard hiervoor is de UML (voor Unied Modeling Language–de naam zegt het al) en in het bijzonder zijn deeltaal OCL (voor Object Constraint Language), die ertoe dient de bedoelde betekenis van constructies in UML nader vast te leggen.

Jammer genoeg is noch de betekenis van UML, noch die van OCL eenduidig vastgelegd. (Sommige bronnen beweren dat dit met opzet gebeurd is om tegenstrijdige industriële belangen te dienen). Het is duidelijk dat als je niet precies weet wat een bepaalde taalkonstructie betekent, je hem ook niet met 100 % zekerheid kunt gebruiken om een apparaat te sturen waar mensenlevens van afhangen.

Om in deze situatie verandering te brengen is dit proefschrift geschreven. Het beschrijft een formele, dat wil zeggen, in wiskundige zin exacte, semantiek voor de taalconstructies van UML en OCL, en voorziet deze talen van een zinvol typesysteem dat ertoe dient om aan te geven in welke context een UML of OCL taalkonstructie zinvol te gebruiken is. Dit type systeem is, als onderdeel van dit proefschrift, geïmplementeerd, zodat het voldoen aan de betreffende typerings regels elektronisch kan worden gecheckt.

Om te bewijzen dat deze semantiek eenduidig is, is hij omgezet in de specificatie-taal van PVS, een elektronisch systeem dat bewijzen van wiskundige stellingen op hun correctheid checkt en dat veel gebruikt wordt om er correctheidsbewijzen van programma’s elektronisch mee te controleren.

Vervolgens worden in dit proefschrift een paar karakteristieke toepassingen van UML korrekt bewezen, waarbij de gebruikte semantiek die is welke in dit proefschrift vastgelegd wordt, de architectuur van deze toepassingen in UML gegeven wordt en hun functionaliteit in OCL wordt gespecificeerd.
Samenvatting

De eerste toepassing betreft een programma voor de Zeef van Eratosthenes, dat ertoe dient de priemgetallen te genereren. Dit ontleent zijn belang aan het feit dat de desbetreffende “zeef” zich in principe een onbegrensd aantal malen (recursief) oproepen kan. Er wordt aangegeven hoe dit probleem in PVS gecodeerd kan worden, waarna de correctheid van dit programma met behulp van PVS bewezen wordt.

De tweede toepassing is ontleend aan een programma dat gebruikt wordt door de Koninklijke Luchtmacht in hun verkenningsvliegtuigen om daar zeer nauwkeurige fotos mee te maken. Wanneer namelijk vanuit straaljagers gefotografeerd wordt, moet voor nauwkeurige fotos een compensatie-mechanisme ingebouwd worden in verband met de tijdens een opname afgelegde afstand; die moet door bewegende spiegels gecompenseerd worden. Van het centrale deel van het elektronische ingebouwde real-time systeem dat de beweging van deze spiegels regelt wordt een nauwkeurige specificatie in OCL gegeven en met behulp van PVS bewezen dat de UML beschrijving van de architectuur van het desbetreffende besturingssysteem aan deze specificatie voldoet.

Daarmee wordt aangetoond dat deze semantieken en hun omzetting in PVS zich er in principe toe lenen om er industriële toepassingen, waarvan architectuur en functionaliteit in UML en OCL beschreven zijn, mee korrekt te bewijzen.
Curriculum Vitæ

January 30, 1975  Born in Pinneberg, Germany.


Minor subject: electrical engineering.


April 1999–December 2000  Student assistant at CAU, Institute of Computer Science and Applied Mathematics (Software Technology), implementing static analysers for sequential function charts.

October 2000–December 2001  Assistant professor (nebenamtlicher Dozent) lecturing on Algorithms and data structures at FH Nordakademie.

January 2001–today  Researcher at Christian-Albrechts-Universität zu Kiel, working for the IST-project Omega (IST-2001-33522), DFG/NWO-project Mobi-J (RO-1122/9-1 and RO1122/9-2), and DFG-project SFC-Check (LA-1021/6-1).

Current address:  Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
24098 Kiel
Germany
Titles in the IPA Dissertation Series

Titles in the IPA Dissertation Series are not available from Lehmanns Media. Please contact the IPA Secretariat (http://www.win.tue.nl/ipa/) for help on obtaining a dissertation from this list.


A.M. Geerling. Transformational Development of Data-Parallel Algorithms. Faculty of Mathematics and Computer Science, KUN. 1996-02

P.M. Achten. Interactive Functional Programs: Models, Methods, and Implementation. Faculty of Mathematics and Computer Science, KUN. 1996-03

M.G.A. Verhoeven. Parallel Local Search. Faculty of Mathematics and Computing Science, TUE. 1996-04

M.H.G.K. Kesseler. The Implementation of Functional Languages on Parallel Machines with Distrib. Memory. Faculty of Mathematics and Computer Science, KUN. 1996-05


H. Doornbos. Reductivity Arguments and Program Construction. Faculty of Mathematics and Computing Science, TUE. 1996-08

D. Turi. Functorial Operational Semantics and its Denotational Dual. Faculty of Mathematics and Computer Science, VUA. 1996-09


N.W.A. Arends. A Systems Engineering Specification Formalism. Faculty of Mechanical Engineering, TUE. 1996-11

P. Severi de Santiago. Normalisation in Lambda Calculus and its Relation to Type Inference. Faculty of Mathematics and Computing Science, TUE. 1996-12


M.M. Bonsangue. Topological Dualities in Semantics. Faculty of Mathematics and Computer Science, VUA. 1996-14

B.L.E. de Fluitert. Algorithms for Graphs of Small Treewidth. Faculty of Mathematics and Computer Science, UU. 1997-01

W.T.M. Kars. Process-algebraic Transformations in Context. Faculty of Computer Science, UT. 1997-02


C.J. Bloo. Preservation of Termination for Explicit Substitution. Faculty of Mathematics and Computing Science, TUE. 1997-05


F.A.M. van den Beuken. A Functional Approach to Syntax and Typing. Faculty of Mathematics and Informatics, KUN. 1997-07

A.W. Heerink. Ins and Outs in Refusal Testing. Faculty of Computer Science, UT. 1998-01

G. Naumoski and W. Alberts. A Discrete-Event Simulator for Systems Engineering. Faculty of Mechanical Engineering, TUE. 1998-02

J. Verriet. Scheduling with Communication for Multiprocessor Computation. Faculty of Mathematics and Computer Science, UU. 1998-03


E. Voermans. Inductive Datatypes with Laws and Subtyping – A Relational Model. Faculty of Mathematics and Computing Science, TUE. 1999-01
H. ter Doest. **Towards Probabilistic Unification-based Parsing.** Faculty of Computer Science, UT. 1999-02

J.P.L. Segers. **Algorithms for the Simulation of Surface Processes.** Faculty of Mathematics and Computer Science, TUE. 1999-03

C.H.M. van Kemenade. **Recombinative Evolutionary Search.** Faculty of Mathematics and Natural Sciences, UL. 1999-04

E.I. Barakova. **Learning Reliability: a Study on Indecisiveness in Sample Selection.** Faculty of Mathematics and Natural Sciences, RUG. 1999-05

M.P. Bodlaender. **Scheduler Optimization in Real-Time Distributed Databases.** Faculty of Mathematics and Computer Science, TUE. 1999-06

M.A. Reniers. **Message Sequence Chart: Syntax and Semantics.** Faculty of Computer Science, UT. 1999-06

J.P. Warners. **Nonlinear approaches to satisfiability problems.** Faculty of Mathematics and Computer Science, TUE. 1999-08

J.M.T. Romijn. **Analysing Industrial Protocols with Formal Methods.** Faculty of Computer Science, UT. 1999-09

P.R. D’Argenio. **Algebras and Automata for Timed and Stochastic Systems.** Faculty of Computer Science, UT. 1999-10

G. Fábián. **A Language and Simulator for Hybrid Systems.** Faculty of Mechanical Engineering, TUE. 1999-11

J. Zwanenburg. **Object-Oriented Concepts and Proof Rules.** Faculty of Mathematics and Computer Science, TUE. 1999-12

R.S. Venema. **Aspects of an Integrated Neural Prediction System.** Faculty of Mathematics and Natural Sciences, RUG. 1999-13

J. Saraiva. **A Purely Functional Implementation of Attribute Grammars.** Faculty of Mathematics and Computer Science, UU. 1999-14

R. Schiefer. **Viper, A Visualisation Tool for Parallel Program Construction.** Faculty of Mathematics and Computer Science, TUE. 1999-15

K.M.M. de Leeuw. **Cryptology and Statecraft in the Dutch Republic.** Faculty of Mathematics and Computer Science, UvA. 2000-01

T.E.J. Vos. **UNITY in Diversity.** A stratified approach to the verification of distributed algorithms. Faculty of Mathematics and Computer Science, UU. 2000-02

W. Mallon. **Theories and Tools for the Design of Delay-Insensitive Communicating Processes.** Faculty of Mathematics and Natural Sciences, RUG. 2000-03

W.O.D. Griffioen. **Studies in Computer Aided Verification of Protocols.** Faculty of Science, KUN. 2000-04

P.H.F.M. Verhoeven. **The Design of the MathSpad Editor.** Faculty of Mathematics and Computing Science, TUE. 2000-05

J. Fey. **Design of a Fruit Juice Blending and Packaging Plant.** Faculty of Mechanical Engineering, TUE. 2000-06

M. Franssen. **Cocktail: A Tool for Deriving Correct Programs.** Faculty of Mathematics and Computing Science, TUE. 2000-07

P.A. Olivier. **A Framework for Debugging Heterogeneous Applications.** Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2000-08

E. Saaman. **Another Formal Specification Language.** Faculty of Mathematics and Natural Sciences, RUG. 2000-10

M. Jelasity. **The Shape of Evolutionary Search Discovering and Representing Search Space Structure.** Faculty of Mathematics and Natural Sciences, UL. 2001-01

R. Ahn. **Agents, Objects and Events a computational approach to knowledge, observation and communication.** Faculty of Mathematics and Computer Science, TU/e. 2001-02

M. Huisman. **Reasoning about Java programs in higher order logic using PVS and Isabelle.** Faculty of Science, KUN. 2001-03

I.M.M.J. Reymen. **Improving Design Processes through Structured Reflection.** Faculty of Mathematics and Computer Science, TU/e. 2001-04

M. Blom. **Term Graph Rewriting: syntax and semantics.** Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2001-05

R. van Liere. **Studies in Interactive Visualization.** Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2001-06

J. Hage. Structural Aspects of Switching Classes. Faculty of Mathematics and Natural Sciences, UL. 2001-08

M.H. Lamers. Neural Networks for Analysis of Data in Environmental Epidemiology: A Case-study into Acute Effects of Air Pollution Episodes. Faculty of Mathematics and Natural Sciences, UL. 2001-09

T.C. Ruys. Towards Effective Model Checking. Faculty of Computer Science, UT. 2001-10

D. Chkliaev. Mechanical verification of concurrency control and recovery protocols. Faculty of Mathematics and Computing Science, TU/e. 2001-11


D. Bošnački. Enhancing state space reduction techniques for model checking. Faculty of Mathematics and Computing Science, TU/e. 2001-14

M.C. van Wezel. Neural Networks for Intelligent Data Analysis: theoretical and experimental aspects. Faculty of Mathematics and Natural Sciences, UL. 2002-01

V. Bos and J.J.T. Kleijn. Formal Specification and Analysis of Industrial Systems. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2002-02

T. Kuipers. Techniques for Understanding Legacy Software Systems. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-03


N. van Vugt. Models of Molecular Computing. Faculty of Mathematics and Natural Sciences, UL. 2002-07

A. Fehnker. Citius, Vilius, Melius: Guiding and Cost-Optimality in Model Checking of Timed and Hybrid Systems. Faculty of Science, Mathematics and Computer Science, KUN. 2002-08

R. van Stee. On-line Scheduling and Bin Packing. Faculty of Mathematics and Natural Sciences, UL. 2002-09


M.B. van der Zwaag. Models and Logics for Process Algebra. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-11


J.J. van Hemert. Applying Evolutionary Computation to Constraint Satisfaction and Data Mining. Faculty of Mathematics and Natural Sciences, UL. 2002-14


Y.S. Usenko. Linearization in µCRL. Faculty of Mathematics and Computer Science, TU/e. 2002-16

J.J.D. Aerts. Random Redundant Storage for Video on Demand. Faculty of Mathematics and Computer Science, TU/e. 2003-01

M. de Jonge. To Reuse or To Be Reused: Techniques for component composition and construction. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2003-02

S.M. Bohte. *Spiking Neural Networks*. Faculty of Mathematics and Natural Sciences, UL. 2003-04


S.V. Nedea. *Analysis and Simulations of Catalytic Reactions*. Faculty of Mathematics and Computer Science, TU/e. 2003-06


M.H. ter Beek. *Team Automata – A Formal Approach to the Modeling of Collaboration Between System Components*. Faculty of Mathematics and Natural Sciences, UL. 2003-10


G.I. Jojgov. *Incomplete Proofs and Terms and Their Use in Interactive Theorem Proving*. Faculty of Mathematics and Computer Science, TU/e. 2004-02

P. Frisco. *Theory of Molecular Computing – Splicing and Membrane systems*. Faculty of Mathematics and Natural Sciences, UL. 2004-03

S. Maneth. *Models of Tree Translation*. Faculty of Mathematics and Natural Sciences, UL. 2004-04

Y. Qian. *Data Synchronization and Browsing for Home Environments*. Faculty of Mathematics and Computer Science and Faculty of Industrial Design, TU/e. 2004-05


L. Cruz-Filipe. *Constructive Real Analysis: a Type-Theoretical Formalization and Applications*. Faculty of Science, Mathematics and Computer Science, KUN. 2004-07

E.H. Gerding. *Autonomous Agents in Bargaining Games: An Evolutionary Investigation of Fundamentals, Strategies, and Business Applications*. Faculty of Technology Management, TU/e. 2004-08


J. Pang. *Formal Verification of Distributed Systems*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-14

F. Alkemade. *Evolutionary Agent-Based Economics*. Faculty of Technology Management, TU/e. 2004-15

E.O. Dijk. *Indoor Ultrasonic Position Estimation Using a Single Base Station*. Faculty of Mathematics and Computer Science, TU/e. 2004-16

S.M. Orzan. *On Distributed Verification and Verified Distribution*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-17


P.J.L. Cuijpers. *Hybrid Process Algebra*. Faculty of Mathematics and Computer Science, TU/e. 2004-20

N.J.M. van den Nieuwelaar. *Supervisory Machine Control by Predictive-Reactive Scheduling*. Faculty of Mechanical Engineering, TU/e. 2004-21


R. Ruimerman. *Modeling and Remodeling in Bone Tissue*. Faculty of Biomedical Engineering, TU/e. 2005-02


H.M.A. van Beek. *Specification and Analysis of Internet Applications*. Faculty of Mathematics and Computer Science, TU/e. 2005-05


I. Kurtev. *Adaptability of Model Transformations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-08

T. Wolle. *Computational Aspects of Treewidth - Lower Bounds and Network Reliability*. Faculty of Science, TU/e. 2005-09


A.M.L. Liekens. *Evolution of Finite Populations in Dynamic Environments*. Faculty of Biomedical Engineering, TU/e. 2005-11

J. Eggermont. *Data Mining using Genetic Programming: Classification and Symbolic Regression*. Faculty of Mathematics and Natural Sciences, UL. 2005-12

B.J. Heeren. *Top Quality Type Error Messages*. Faculty of Science, UU. 2005-13

G.F. Frehse. *Compositional Verification of Hybrid Systems using Simulation Relations*. Faculty of Science, Mathematics and Computer Science, RU. 2005-14


P. Zoeteweij. *Composing Constraint Solvers*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-18


M. Valero Espada. *Modal Abstraction and Replication of Processes with Data*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2005-20

A. Dijkstra. *Stepping through Haskell*. Faculty of Science, UU. 2005-21


M. Kyas. *Verifying OCL Specifications of UML Models: Tool Support and Compositionality*. Faculty of Mathematics and Natural Sciences, UL. 2006-05