Abstracts Collection
Workshop Trustworthy Software 2006
INTERREG IIIC/e-Bird

Serge Autexier, Stephan Merz, Leon van der Torre,
Reinhard Wilhelm and Pierre Wolper

Abstract. On 18-19 May 2006, the Saarland University organized a two-day workshop about "Trustworthy Software" in order to present and foster the research competence in the SaarLorLuxWallonie region in the area of developing safe, secure and reliable software, computers and networks. As part of the Interreg III C E-Bird project "Recherches sans frontières/Forschen ohne Grenzen" it provided an excellent forum especially for young scientists to present and discuss recent results, new ideas and future research directions to a transnational audience from the SaarLorLuxWallonie region. The workshop consisted of 21 regular presentations and one invited talk. Abstracts of all presentations are collected in this paper, including links to extended abstracts or full papers. The first section directs to the preface of the proceedings.

Keywords. Software evolution, Modularity, Automated debugging, Dependability assurance, Failure analysis, Static program analysis, Infinite and Finite-state verification, Runtime verification, Theorem proving, Access control, Security analysis, Security protocols, E-Voting

Preface – Workshop Trusworthy Software 2006

Serge Autexier (DFKI - Saarbrücken, D)

As part of the Interreg III C/E-Bird project "Recherches sans frontières/Forschen ohne Grenzen" the Saarland University organized a two-day workshop about "Trustworthy Software" in order to present and foster the research competence in the SaarLorLuxWallonie region in the area of developing safe, secure and reliable software, computers and networks. The workshop especially provided a forum for young scientists to present their research to a transnational audience from the SaarLorLuxWallonie region and consisted of 21 regular presentations and one invited presentation.

Keywords: Trustworthy software, preface

Joint work of: Autexier, Serge; Merz, Stephan; van der Torre, Leon; Wilhelm, Reinhard; Wolper, Pierre

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2006/693

Superposition Based Verification of Invariants. Application to Parameterized Systems.

Jean-François Couchot (Univeristé de Franche-Comté, F)

The harvey theorem prover implements a decision procedure for ground first order equational formulae in the array theory. This work provides valuable insights into the applicability of such a prover for the verification of safety properties expressed by an invariant on parameterized systems.

The soundness of such parameterized programs has to be checked uniformly, i.e. once for all its sizes. Such programs can be verified by a deductive fix point calculus whose proof obligations are discharged into a prover that allows quantified formulae.

Initiated by Graf and Saïdi who discharged their evolution conditions into the PVS prover, many studies have concerned the systems based on linear arithmetic constraints after a convenient counting abstraction.

We suggest a more basic but unifying approach in which the parameter ranges over a finite set. We show that such a framework is adequate for industrial test cases and uniform distributed systems. The specifications are written with the set theoretical B machine notation and we exploit an existing weakest precondition calculus for this method. Then, we introduce an invariant strengthening calculus, obtained as the refinement of a trivial calculus. We provide different methods for translating the evolution condition of this calculus into some equational logics. Their main objective is to make harvey discharge them as fast as possible.

On an industrial scale example, we show that this approach is more efficient than the Atelier B deductive system. Theoretically, we prove the evolution condition decidability, where the framework is some uniform distributed among broadcast and rendez-vous synchronization.

Keywords: Superposition, Verification, Parameterized Systems

SANA - Security Analysis in Internet Traffic through Artificial Immune Systems

Michael Hilker (University of Luxembourg, L)

The Attacks done by Viruses, Worms, Hackers, etc. are a Network Security-Problem in many Organisations. Current Intrusion Detection Systems have significant Disadvantages, e.g. the need of plenty of Computational Power or the Local Installation. Therefore, we introduce a novel Framework for Network Security which is called SANA. SANA contains an artificial Immune System with artificial Cells which perform certain Tasks in order to to support existing systems to better secure the Network against Intrusions. The Advantages of SANA are that it is efficient, adaptive, autonomous, and massively-distributed. In this Article, we describe the Architecture of the artificial Immune System and the Functionality of the Components. We explain briefly the Implementation and discuss Results.

Keywords: Artificial Immune Systems, Network Security, Intrusion Detection, Artificial Cell Communication, Biological-Inspired Computing, Complex Adaptive Systems

Joint work of: Hilker, Michael; Schommer, Christoph

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/694

An Operator-based Approach to Incremental Development of Conform Protocol State Machines

Arnaud Lanoix (LORIA, F)

An incremental development framework which supports a conform construction of Protocol State Machines (PSMs) is presented. We capture design concepts and strategies of PSM construction by sequentially applying some development operators: each operator makes evolve the current PSM to another one. To ensure a conform construction, we introduce three conformance relations, inspired by the specification refinement and specification matchings supported by formal methods. Conformance relations preserve some global behavioral properties. Our purpose is illustrated by some development steps of the card service interface of an electronic purse: for each step, we introduce the idea of the development, we propose an operator and we give the new specification state obtained by the application of this operator and the property of this state relatively to the previous one in terms of conformance relation.

Keywords: Protocol state machine, incremental development, development operator, exact conformance, plugin conformance, partial conformance

Joint work of: Lanoix, Arnaud; Okalas Ossami, Dieu-donné; Souquières, Jeanine

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/695

An Introduction to the Tool Ticc

Axel Legay (University of Liège, B)

This paper is a tutorial introduction to the sociable interface model of [?] and its underlying tool Ticc [?]. The paper starts with a survey of the theory of interfaces and then introduces the sociable interface model that is a game-based model with rich communication primitives to facilitate the modeling of software and distributed systems. The model and its main features are then intensivelly discussed and illustrated using the tool Ticc.

Keywords: Open system, game, interface automata

Joint work of: Legay, Axel; de Alfaro, Luca; Faella, Marco

Isolating Intrusions by Automatic Experiments

Stephan Neuhaus (Universität des Saarlandes, D)

When dealing with malware infections, one of the first tasks is to find the processes that were involved in the attack. We introduce Malfor, a system that isolates those processes automatically. In contrast to other methods that help analyze attacks, Malfor works by experiments: first, we record the interaction of the system under attack; after the intrusion has been detected, we replay the recorded events in slightly different configurations to see which processes were relevant for the intrusion. This approach has three advantages over deductive approaches: first, the processes that are thus found have been experimentally shown to be relevant for the attack; second, the amount of evidence that must then be analyzed to find the attack vector is greatly reduced; and third, Malfor itself cannot make wrong deductions. In a first experiment, Malfor was able to extract the three processes responsible for an attack from 32 candidates in about six minutes.

Keywords: Intrusion Analysis, Malware, Experimentation

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2006/696

Formal Validation of Pattern Matching code

Antoine Reilles (CNRS & LORIA, F)

When addressing the formal validation of generated software, two main alternatives consist either to prove the correctness of compilers or to directly validate the generated code. Here, we focus on directly proving the correctness of compiled code issued from powerful pattern matching constructions typical of ML like languages or rewrite based languages such as ELAN, MAUDE or Tom.

In this context, our first contribution is to define a general framework for anchoring algebraic pattern-matching capabilities in existing languages like C, Java or ML. Then, using a just enough powerful intermediate language, we formalize the behavior of compiled code and define the correctness of compiled code with respect to pattern-matching behavior. This allows us to prove the equivalence of compiled code correctness with a generic first-order proposition whose proof could be achieved via a proof assistant or an automated theorem prover. We then extend these results to the multi-match situation characteristic of the ML like languages.

The whole approach has been implemented on top of the Tom compiler and used to validate the syntactic matching code of the Tom compiler itself.

Keywords: Correctness proofs, compilers, pattern matching, validation

Joint work of: Kirchner, Claude; Moreau, Pierre-Etienne; Reilles, Antoine

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/697

Shape Analysis of Sets

Jan Reineke (Universität des Saarlandes, D)

Shape Analysis is concerned with determining "shape invariants", i.e. structural properties of the heap, for programs that manipulate pointers and heap-allocated storage. Recently, very precise shape analysis algorithms have been developed that are able to prove the partial correctness of heap-manipulating programs. We explore the use of shape analysis to analyze abstract data types (ADTs). The ADT Set shall serve as an example, as it is widely used and can be found in most of the major data type libraries, like STL, the Java API, or LEDA. We formalize our notion of the ADT Set by algebraic specification. Two prototypical C set implementations are presented, one based on lists, the other on trees. We instantiate a parametric shape analysis framework to generate analyses that are able to prove the compliance of the two implementations to their specification.

Keywords: Shape analysis, adt, algebraic specification, invariants, verification, set implementations, imperative programs

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/698

Using Abstraction in Modular Verification of Synchronous Adaptive Systems

Ina Schaefer (TU Kaiserslautern, D)

Self-adaptive embedded systems autonomously adapt to changing environment conditions to improve their functionality and to increase their dependability by downgrading functionality in case of fail- ures. However, adaptation behaviour of embedded systems significantly complicates system design and poses new challenges for guaranteeing system correctness, in particular vital in the automotive domain. Formal verification as applied in safety-critical applications must therefore be able to address not only temporal and functional properties, but also dynamic adaptation according to external and internal stimuli.

In this paper, we introduce a formal semantic-based framework to model, specify and verify the functional and the adaptation behaviour of syn- chronous adaptive systems. The modelling separates functional and adap- tive behaviour to reduce the design complexity and to enable modular reasoning about both aspects independently as well as in combination.

By an example, we show how to use this framework in order to verify properties of synchronous adaptive systems. Modular reasoning in com- bination with abstraction mechanisms makes automatic model checking efficiently applicable.

Keywords: Dependable Embedded Systems, Self-Adaptation, Abstraction, Modular Verification

Joint work of: Schaefer, Ina; Poetzsch-Heffter, Arnd

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/699

Formalizing On Chip Communications in a Functional Style

Julien Schmaltz (Universität des Saarlandes, D)

This paper presents a formal model for representing *any* on-chip communication architecture.

This model is described mathematically by a function, named *GeNoC*. The correctness of *GeNoC* is expressed as a theorem, which states that messages emitted on the architecture reach their expected destination without modification of their content. The model identifies the key constituents common to *all* communication architectures and their essential properties, from which the proof of the *GeNoC* theorem is deduced. Each constituent is represented by a function which has no *explicit* definition but is constrained to satisfy the essential properties. Thus, the validation of a *particular* architecture is reduced to the proof that its concrete definition satisfies the essential properties. In practice, the model has been defined in the logic of the ACL2 theorem proving system.

We define a methodology that yields a systematic approach to the validation of communication architectures at a high level of abstraction. To validate our approach, we exhibit several architectures that constitute concrete instances of the generic model *GeNoC*. Some of these applications come from industrial designs, such as the AMBA AHB bus or the Octagon network from ST Microelectronics.

Keywords: SoC's, NoC's, communication architectures, formal methods, automated theorem proving

Joint work of: Schmaltz, Julien; Borrione, Dominique

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/700

Separation Logic for General Storage

Jan Schwinghammer (Universität des Saarlandes, D)

Separation Logic is a substructural logic that facilitates local reasoning for imperative programs, in the sense that only the reachable part of the store must be taken into account for the verification of a command. In past work, Separation Logic has been developed for heaps containing records of basic data types.

Languages like C and ML, however, are less constrained and permit also the use of code pointers and higher-order references, respectively. The corresponding heap model is commonly referred to as "general storage" (or "higher-order store") since heaps may contain commands.

In this talk I will report on recent joint work with Bernhard Reus, where we make Separation Logic and the benefits of local reasoning available to languages with general storage.

Keywords: Program verification, Separation Logic, higher-order store

Explaining Data Type Reduction in the Shape Analysis Framework

Björn Wachter (Universität des Saarlandes, D)

Automatic formal verification of systems composed of a large or even unbounded number of components is difficult as the state space of these systems is prohibitively large. Abstraction techniques automatically construct finite approximations of infinite-state systems from which safe information about the original system can be inferred. We study two abstraction techniques shape analysis, a technique from program analysis, and data type reduction, originating from model checking. Until recently we did not properly understand how shape analysis and data type reduction relate. In this talk, we shed light on this relation in a comprehensive way. This is a step towards a more unified view of abstraction employed in the static analysis and model checking community.

Keywords: Canonical abstraction, data type reduction, model checking, parameterized system, infinite-state

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2006/701

Full Paper:
http://rw4.cs.uni-sb.de/∼bwachter/thesis.pdf

Relating two standard notions of secrecy

Eugen Zalinescu (UHP & LORIA & INRIA Project CASSIS, F)

Two styles of definitions are usually considered to express that a security protocol preserves the confidentiality of a data s. Reachability-based secrecy means that s should never be disclosed while equivalence-based secrecy states that two executions of a protocol with distinct instances for s should be indistinguishable to an attacker. Although the second formulation ensures a higher level of security and is closer to cryptographic notions of secrecy, decidability results and automatic tools have mainly focused on the first definition so far.

This paper initiates a systematic investigation of situations where syntactic secrecy entails strong secrecy.

We show that in the passive case, reachability-based secrecy actually implies equivalence-based secrecy for signatures, symmetric and asymmetric encryption provided that the primitives are probabilistic. For active adversaries in the case of symmetric encryption, we provide sufficient (and rather tight) conditions on the protocol for this implication to hold.

Keywords: Verification, security protocols, secrecy, applied-pi calculus

Joint work of: Zalinescu, Eugen; Cortier, Véronique; Rusinowitch, Michaël

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/691

Full Paper:
http://www.inria.fr/rrrt/rr-5908.html