Draft Fourth VDM-Overture Workshop Programme

The 26th May 2008, Co-located with FM 08, Turku, Finland

9:00 – 9:10		
Participant	Title	Abstract
Peter Gorm Larsen and Shin Sahara	Welcome and Introduction	Overview of the events that have led to the start of Overture. The current status of the project will be presented and a list of "open issues" will be introduced. These issues will form the basis of the brainstorm session held later. New issues may, of course, be added as the day progresses.

9:10 – 9:35		
Participant	Title	Abstract
Zhiming Liu and **Volker Stolz**	The rCOS Method in Nutshell	The state of the practice in model driven development is the UML-like multi-view and multi-notational modeling and design. However, a major challenge in the practice of model-based software development is to ensure correctness and dependability of the software product. To deal with this challenge, we need a common semantics for the multi-view modeling approach. In this paper, we discuss how the calculus of refinement of component and object systems, rCOS, can be used as such a common semantic model. The discussion focuses on how the main ideas, and techniques of model driven development can be addressed in the framework of rCOS, and how the problems there can be solved. The presentation is informal without mathematical formalization.

9:35: - 10:00		
Participant	Title	Abstract
Zhenbang Chen, Zhiming Liu, **Volker Stolz**	The rCOS tool	The goal of the rCOS tool is to harness state of the art techniques from use case- and model driven-development of component-based systems on top of UML. It facilitates both the development process and the persistence of formal verification artefacts in the model: use cases are specified in multiple views, using Class, Sequence and State Diagrams to capture their behaviour. Functionality is specified in rCOS pre-/postconditions based on UTP. Provably correct refinement steps transform the Requirements Model into a component-based Design Model. Consistency of the different views is ensured, for example by

		checking that the State Diagram accepts the protocol specified in the Sequence Diagram. For component composition, protocol compatibility is verified. To that end, either third party tools like the FDR2 model checker are invoked, or annotations to code skeletons for additional tools like the Java Modeling Language (JML) are generated. The tool is implemented on top of the Eclipse platform using a UML profile, ensuring compatibility with other UML-based software engineering tools.

10:00 – 10:30		
Participant	Title	Abstract
Kristian Bisgaard Lassen and **Simon Tjell**	Developing Tool Support for Problem Diagrams with CPN and VDM++	In this paper, we describe ongoing work on the development of tool support for formal description of domains found in Problem Diagrams. The purpose of the tool is to handle the generation of a CPN model based on a collection of Problem Diagrams. The Problem Diagrams are used for representing the structure and parallel decomposition of a software development problem while the CPN model is used for formal specification of assumed and desired behaviour of the domains found in the Problem Diagrams. After generation, the CPN model will be manually refined and during this process, it is repeatedly validated against structural constraints found in the Problem Diagrams. The generation and validation algorithms as well as the definitions of the two modelling formalisms are specified using VDM++.

10:30 – 11:00 Coffee break

11:00 – 11:30		
Participant	Title	Abstract
Carlos M. G. Vilhena	Overture: Connecting VDM++ and JML	This paper discusses a number of possibilities for automatic conversion between VDM++ and JML, in both directions, as part of a project to enable VDM++ as a front-end for contract-based programming and the possible usage of tool support both from VDM++ and JML. In particular, the project aims at identifying the notational subsets for which the envisaged automatic translation is possible, as well as describing in detail all the limitations encountered. The development of a prototype proof-of-concept implementation for this bi-directional conversion is being carried through. At a latter stage this prototype will be integrated on top

		of the Eclipse platform as part of the Overture Tool.

11:30 – 12:00		
Participant	Title	Abstract
Adriana Sucena Santos	Overture: Combinatorial Test Automation Support for VDM++	Testing is an important, expensive, repetitive and exhaustive task in software development. It does not guarantee that a model has no errors, but the developer can be more confident that a model is working properly after testing it. If the testing process is done along the development, it will have to be frequently repeated because a small change in the model might influence its behaviour. Repeating tests, without automation support, each time a change is made is a tedious manual task. It is also time consuming and, consequently, expensive. This paper suggests a test automation support for the Overture which may be useful for VDM++ developers.

12:00 – 12:30		
Participant	Title	Abstract
Miguel Ferreira and Samuel Silva	Verifying Intel's Flash File System Core	This contribution will report on the use of Alloy and HOL to validate and verify a VDM model of Intel's Flash File System Core specification. The approach uses the VDMTools proof obligation generator and the VDM to HOL translator developed by Sander Vermolen. The VDM to Alloy conversion is manual. In this "all-in-one" approach, modeling and testing takes place in the VDM phase. Alloy is particularly helpful in finding counter examples to proof obligations. The prospect of using the point-free transform as a means to simplify the proof obligations submitted to Alloy (for model checking) and HOL (for theorem proving) is also considered.

12:30 – 14:00 Lunch break

14:00 – 14:30		
Participant	Title	Abstract
Hugo Macedo, John Fitzgerald and Peter Gorm Larsen	Incremental Development of a Distributed Real-Time Model of a Cardiac Pacing System using VDM	The construction of formal models of real-time distributed systems is a considerable practical challenge. We propose and illustrate a pragmatic incremental approach in which detail is progressively added to abstract system-level specifications of functional and timing properties via intermediate models that express system architecture, concurrency and timing behaviour. The approach is illustrated by developing a new formal model of the cardiac

		pacemaker system proposed as a “grand challenge” problem in 2007. The models are expressed using the Vienna Development Method (VDM) and are validated primarily by scenario-based tests, including the analysis of timed traces. We argue that the insight gained using this staged modelling approach will be valuable in the subsequent development of implementations, and in detecting potential bottlenecks within suggested implementation architectures.

14:30 – 15:00		
Participant	Title	Abstract
Sander Vermolen	Automating Consistency Proofs of VDM++ Models using HOL	The value of formal models depends upon their consistency and the features available to prove consistency. Hence, it is important to have access to efficient proof support which is able to automate a large part of the consistency proof. We have developed a tool that can perform an automatic translation of a large subset of VDM++ and its associate proof obligations, which ensure model consistency, to the theorem prover HOL. In addition, powerful tactics have been constructed to discard most of the proof obligations automatically. The application of our approach to a number of case studies shows that a high degree of automation can been achieved.

15:00 – 15:30		
Participant	Title	Abstract
Marcel Verhoef	Co-simulation of Distributed Embedded Real-time Control Systems – Coupling VDM++ to 20-sim	The complexity of contemporary real-time embedded control systems is increasing continuously. This is primarily caused by tighter control objectives, challenging functional and performance demands, cost-price optimizations and the use of novel, mostly distributed, system-on-chip architectures. The challenge of the system architect is to get this increased complexity under control as soon as possible, preferably using rigorous engineering approaches. Although domain specific design methods and tools are performing better, the real challenge in industry is to close the well-known inter-disciplinary design gap. The author has worked on the semantic integration of two formal techniques: VDM++ and 20-sim, to address this issue. The former method is well known from computer science, the later is well-established in control engineering. In this talk, we focus on the

		design of the tool support for this approach and how it can be used in a practical engineering situation. As a case study, we use the development of a paper path of a high-end office printer to illustrate the impact of the combined solution.

15:30 – 16:00 Coffee break

16:00 – 16:30		
Participant	Title	Abstract
Shin Sahara	VDMTools Past and Future Plans	

16:30 – 17:30		
Participant	Title	Abstract
All	Brainstorm session	The closing session of the day will be an open format brainstorm session, intended to make progress with a number of selected issues relating to Overture. Here we hope to have input from Colin Snook from the DEPLOY project with whom we can discuss how best to share experience and sources for Eclipse plug-ins.

19:00 Workshop dinner (pay your own food and drinks)