

**Overture Workshop
Programme
Monday 18 July 2005
Newcastle upon Tyne, UK
Co-located with FM'05**

9:00 – 9:20		
<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Nico Plat/Peter Gorm Larsen	Introduction to Overture	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.

9:20 – 9:35		
<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Shin Sahara	Update on VDMTools	VDMTools is a well-accepted, industrial tool that fully supports VDM++ and its development cycle. This talk will address recent developments and will elaborate on the relation with Overture.

9:35 – 10:15		
<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Jacob Porsborg Nielsen	Designing a Flexible Kernel providing VDM++ Support for Eclipse	A kernel supporting VDM++ development for Eclipse has been designed and implemented. The focus of this project was to exploit the possibilities of the Eclipse framework to create a flexible and easy extendable kernel. This includes an editor capable of: parsing a VDM++ specification to an abstract syntax tree (AST), exporting an AST to XML, constructing an AST from an XML file, and generating a plain text representation to the editor from an AST. The parser does syntax analysis and provides error-handling mechanisms. All these facilities are integrated with Eclipse. The possibilities of the kernel will be presented and an overview will be given of how other developers can extend it with e.g. type checking or code generation capabilities.

10:15 – 10:30: Coffee break

10:30 – 11:15		
<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Joost Visser	Grammar-centered Development of VDM Support	Starting from the ISO language reference we have developed an industrial strength grammar for the VDM specification language. We present both the development process, and its result. The employed methodology can be described as iterative grammar engineering and includes the application of techniques such as grammar metrication, unit testing, and test coverage analysis. The result is a VDM grammar of industrial strength, in the sense that it is well-tested, it can be used for fast parsing of high volumes of VDM specifications, and it allows automatic generation of support for syntax tree representation, traversal, and interchange. In particular, we have generated Haskell support for parsing VDM, traversing the resulting ASTs, representing the ASTs in XML and in the ATerm maximal sharing interchange format, and for pretty-printing the ASTs back to VDM's surface syntax. This front-end has proven its usefulness in the implementation of VooDooM, a tool that supports generation of relational models from VDM data type specification.

11:15 – 12:00		
<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Marcel Verhoef	On The Use of VDM++ for Specifying Real-time Systems	One of the important qualities of formal techniques is that abstraction is used to focus on the core properties of the system design. In particular concurrency and real-time have a long standing tradition in research, but their practical

		<p>application has not yet propagated sufficiently towards Industry. The notations used are often not compatible with traditional design, which poses a hurdle for practical application. Some language extensions have been proposed for VDM++ to facilitate specification of real-time systems, to overcome this hurdle. The purpose of this presentation is to assess these language extensions based on the results of an industrial case study.</p>
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

12:00 – 12:30

<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Joost Visser	Camila Revival: VDM meets Haskell	<p>We have experimented with modelling some of the key concepts of the VDM specification language inside the functional programming language Haskell. For instance, VDM's sets and maps are directly available as data types defined in standard libraries; we merely needed to define some additional functions to make the match complete. A bigger challenge is posed by VDM's data type invariants, and pre- and post-conditions. For these we resorted to Haskell's constructor class mechanism, and its support for monads. This allows us to switch between different modes of evaluation (e.g. with or without property checking) by simply coercing user defined functions and operations to different specific types.</p> <p>Historical note: The Camila system http://camila.di.uminho.pt is a "A System for Software Development using Formal Methods" developed and used by the Formal Methods groups from Braga in the early nineties. This locally brewed system was shelved in favour of VDM (for specification) and Haskell (for development) when these gained momentum in the late nineties. Making these two languages meet is the aim of the Camila Revival effort.</p>

12:30 – 14:00: Lunch

14:00 – 14:45

<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Bernhard Aichernig	On the Value of Fault Injection on the Modelling Level	<p>This talk is about recent work on specification-based test case generation via model mutation. The idea is not to cover a specification structurally (e.g. all expressions/statements), but to cover a predefined set of possible faults (like semantic Misunderstandings). A prototype tool has been made that generates such test cases, currently for OCL specifications, that could be easily adapted to VDM++. This could become part of Overture. I would motivate the talk from my experience with test case generation from VDM++ specifications.</p>

14:45 – 15:30

<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
Alexander Petrenko	VDM++ versus Programming Language Extensions	<p>This presentation will address a comparison of VDM++ and specification extension of programming languages for modelling and test generation (JML, Spec#, UniTesK). Different aspects of VDM++ use will be considered in the full cycle of software development. Pros and Cons will be discussed from different points of view: architects, designers, developers, testers, managers, QA staff, etc.</p>

15:30 – 15:45: Coffee break

15:45 – 17:30

<i>Participant</i>	<i>Title</i>	<i>Abstract</i>
All	Brainstorm Session	<p>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.</p>

19:30: Workshop dinner