Notes from Discussions at the 3rd Overture Workshop, Newcastle, December 2006

John Fitzgerald, editing notes by Jeremy Bryans

BKA: Bernhard Aichernig
ZA: Zoe Andrews
JB: Jeremy Bryans
JCB: Juan Bicarregui
JC: Joey Coleman
JSF: John Fitzgerald
JH: John Hughes
CBJ: Cliff Jones
PGL: Peter Gorm Larsen
JNO: José Oliveira
NP: Nico Plat
RP: Richard Payne
SS: Shin Sahara
MV: Marcel Verhoef

These notes are not verbatim. They are edited and regrouped fragments from Jeremy's notes made during the meeting. The aim of the discussion was to identify topics and modes of future research activity for the VDM community

Actions:

1. JSF to convene a semantics group (with BKA, JC, PGL, JNO). Initial topics include VDM++ semantics and proof theory development.
2. PGL to convene a group on tools.
3. BKA & NP to convene a group on methods and applications, looking initially at the JML link.
4. JC to explore VDM/Maude relationship.

Proof

Developments in proof support and model checking are worth close consideration now – somethings are clearly possible that weren't practical in the days of MuRal or perhaps even in PROSPER. Further, SS is asking about future directions to develop VDMTools (test generation, model checking or proof). JNO had presented an approach base on partial relations. Some tool-support (a "Galculator") is under development but at an early stage.

PGL asked How does partial function or partial relation-based reasoning differ from e.g. Mural? CBJ responded that currently we are doing proofs at the point level and need tools for proving at the relation level. Mural needs rebuilding. JH has done some work on this as his 3rd year project. We also need to look at RODIN, especially proof on the fly. JCB suggested that we examine proofs being done "instantly" on Grid infrastructure.

CBJ urged research on how to offer help when the proof doesn't work. We also need potentially huge theory bases (Isabelle has a process for validating theories before they are "approved"). There was the suggestion that we look again at resolution and at first order theorem provers such as Vampire, Spass etc. JCB pointed out that we want true, false and undefined from model checkers. We need falsifiers based on test cases. JCB: Discharging proof obligations can happen straight away with {yes, no, unproven} results. Just knowing which obligation fails is a help.

The Overture Language

BKA asked how large the Overture language should be. PGL: It's less important that things conform to ISO, more important that we do experiments.
CBJ pointed out that languages will die, but genericity reduces the need to rebuild tools. The RODIN tools are built on Eclipse. Originally a generic system, extendible to VDM etc. was envisages. However, good user feedback tends to demand language-specific features. Hence the RODIN tools are now rather specific to Event-B. We should also examine using symbolic execution to debug before proving.

In a discussion on adopting material from other languages like RSL?

- people stick with what they know and the quality of feedback they get on models. Tool support is the decider.
- Test case generation is important (CBJ). DIRC tried combining evidence from generated tests and from proof. You might choose test cases over code analysis for C programs, but code over test for B. Further, maximising reuse after changes is very important.
- Evolution of specifications and rationale are important (JNO, JSF). We need to be able to evolve tests as the specification evolves.

Contributing to the Verified Software Repository

JCB gave an indication of the current state of work on the verified software repository as part of GC6. There was some discussion of contributions from the VDM community.

- We need to know what people do with specifications (CBJ). There are 5 case studies in RODIN, ranging is size up to an air traffic control related study that has about 1000 pages of specification. JCB: We need to make large models available as a first step, then classify and search models.
- Contributions could take many forms: specifications, developments, tools and challenge problems. For example, Peter Mosses is collecting programming language definitions. The Vienna Lab definitions will be scanned and put up. Community efforts like Mondex gain added value from the coordinated efforts of teams to specify them. Any specification of an interesting object can go into the VSR. For example, our pi-calculus model of Mondex just needs a proof that no money is generated: using pi-calculus gave us a good overall view of the system including mobility. Partial specifications are also good for the VSR.
- The next challenge problem is the fault tolerant file store on flash memory for spacecraft based on the Posix standard (CBJ). The fault tolerant model is likely to

require rely conditions. JSF wondered if the VDM "lightweight" simulation-based approach would be worth pursuing.

There was general agreement that we must treat the handling of one of the grand challenge problems as a priority. It's a measure of the success of the GC initiative that we felt it essential to do this or else VDM is effectively dead (JNO). We could take part in the Posix GC, and draw conclusions from that as a means of assessing the current state of the art for VDM and define long-term challenges for the formalism (see roadmapping below) (BKA) It may be advisable to use Mondex first. We have three research streams: semantics, tools and applications. The interdependencies between them should help determine research priorities in each stream. We can envisage research groupings pursuing an agenda in each stream. (JSF)

The list of research streams should be placed on the overture wiki and we should invite people to sign up (PGL).

In tackling Mondex, we may need to highlight the difference between our approach and the others (BKA) but many of the existing attempts are similar in terms of the formalisms applied – there are several model-oriented approaches already – nothing wrong with our adding to the list (JSF). It was noted that the Mondex study emphasises proof; currently we use VDM as the basis for a lightweight FM approach exploiting the interpreter. RAISE also uses an interpreter. Does mondex have undefinedness issues? (JSF, BKA, PGL)

Roadmapping

Other scientific communities have challenges that are part of an agenda updated by leaders in the field. MV suggested that we should develop and maintain a technology roadmap (published on vdmportal.org).

- The PROSPER research is an important building block. What are the others? (PGL)
- Real-time and Concurrency are needed (JSF, CBJ). However we don't need those things to be able to make a contribution to VSR now – we're not obliged to deliver proofs either. We can make a contribution today, stressing our distinctive modelling and interpreter-based approach. JCB added that VSR is a long-term goal, don't need to be able to do everything in order to make a contribution now.
- MV suggested that we should begin VSR contributions by pushing our current technology as far as possible using the current tool set. That will itself raise questions we won't be able to answer.

A 20-year goal

Even then, Posix is a 2 year effort. Looking farther ahead, what is a good 20-year goal? This might induce different questions (CBJ). JCB suggested we consider an equivalent of the large-scale experiments that form a focus of the work of physicists.

- Is Unix a 20-year goal? (JNO) For any OS with a big market or a critical systems market such as air traffic control,. delivering a clear architecture, plus proofs and end user tools would be beneficial.
- What will be good in 10 years? For example, think about Grid-based OS – Linux plus grid middleware. (JCB)
- Important that we consider the whole system (FMs are good at system level description) (PGL, JCB)
- The health service and its associated billing systems are significant areas of current interest, certainly in the UK (CBJ). Legislation, full of invariants and contradictions, is often at the root of such problems (JNO).

Links to Static Analysis Approaches (JML etc)

Concurrency, time and mobility all yield challenging applications. Considering our possible interaction with the JML community, what would a JML user get or want from VDM? What is the value to them of VDM's abstract model? (BKA)

- We need to understand just how and how widely JML is used. (JNO)
- We also need to understand the relationship with ESC, ESC/Java, Spec#.
- VDM may be able to add a level of abstract analysis.
- Contact the JML community through Gary Leavens, Joe Kiniry et al. It's perhaps early to be bringing these people in (NP), but we could develop a list of thought-out relevant issues as a basis for further discussion using vdm-forum (PGL). NP & BKA should focus us: advise what to read and what problems to consider – perhaps have a virtual reading group?

Mobile & Distributed Systems

Design complexity of current mobile systems is hundreds of times bigger than is possible using FMs now. We could target that via a view of abstract systems (MV). Continuous time is also a great challenge (PGL) but just distributed systems are hard enough to analyse. A car contains 80 cpus and 1.5MLoC.

- How about the verified car as a challenge? (JNO)
- How about reconfiguration? We could build a study around component-based systems. There are great possibilities in the network-enabled capability area because NEC weakens constraints. (JSF)

The Semantics Group

Actions in the Semantics group: examine Maude (JC). The subgroup on semantics should probably prioritise on the proof theory, working with the tools group (JSF). We need the work to be driven by projects and mini-projects, for example exploring VDM in Maude (NP,PGL). We need to contact other groups working on tools as well, but include non-tool projects (MV,PGL).

How can we keep the semantics work focussed? Resourcing work is an issue. Students? They will be widely spread rather than focussed in a single group (JSF)(BKA: How did Jim Woodcock do this for Circus?)

Identifying a Kernel VDM++ Language

The semantics of VDM++ is probably necessary for credibility, but not that interesting (JSF). We should focus on a small kernel language first. We (JNO) have data on the use of language constructs in a selection of models. The kernel is relevant to all three proposed streams of work (semantics, methods & applications, tools)(JSF). We can look at other languages, e.g. look at the RAISE subset used for Mondex (BKA). Sander Vermolen will be looking at proof support in the first 6-7 months of 2007 and will likely work to a subset containing implicit operations and functional models (MV)

The Overture tool is a playground for experimenting with changes to the semantics for tool support and proof. We should just build it up, adding features when its safe to do so in proof theory. (BKA, PGL, MV)