

Applying Event-B to Mondex

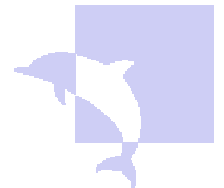
Michael Butler

Divakar Yadav

University of Southampton

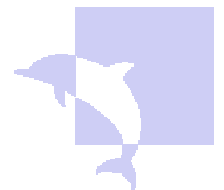
26 May 2006

Mondex Meeting



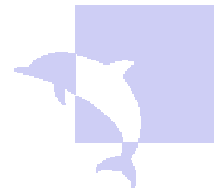
Overview of Plans

- Model and refine using Event B (action systems)
 - Abstract spec describes *Olympian* view of service
 - Refinements introduce design details / environmental assumptions
 - Decomposition to extract architectural components
 - Revise refinement chain where necessary
- Tools:
 - ProB: model checker (Soton + Dusseldorf)
 - B4Free: PO generator and prover (ClearSy)



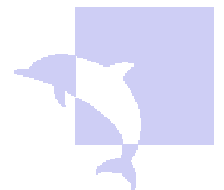
Refinement proof issues

- Formulating the gluing invariant is an iterative process that proceeds hand-in-hand with proof
- **Engineering perspective:** gluing invariant provides insight into *why* a design decision works
- Trade-off between granularity of refinement steps and ease of proof
- Form of invariants:
 - algebraic vs quantifications



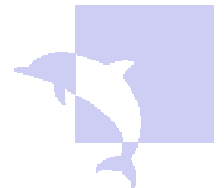
Refinement Chains

- Abstract specification with atomic value transfer (A)
- Detailed design specification with protocol steps and possible failures (D)
- **Goal:** construct a refinement chain with intermediate models to demonstrate that A is refined by D.
- Intermediate models can be viewed as lemmas that factorise the proof into more manageable chunks
 - Leads to higher levels of automatic proof as the invariants are simpler
- Intermediate models increase the transparency of the design rationale



Not top down !

- A and D are (relatively!) fixed but
- Refinement chain will evolve
- Intermediate models will evolve hand in hand with the proof
- Sometimes convenient to introduce new intermediate models
- Typically the degree of automatic proof improves as refinement chain evolves



Overview of refinement chain

- M0

- Atomic transfer of value
- value lost in transactions is recoverable

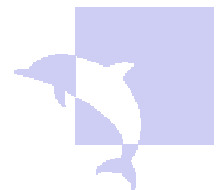
value0 purse →

lost purse →

- Operations

AddPurse, BalanceCheck

TransferOk, TransferFail, Recover



M1: exchange of value split into 2 steps

- Exchange table:

$\text{exchP} \subseteq \text{purse} \times \text{purse}$

$\text{dom}(\text{exchP}) \cap \text{ran}(\text{exchP}) = \{ \}$

$\text{exchV} \subseteq \text{purse} \times \{ \}$

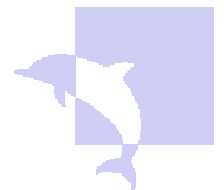
$p \in \text{dom}(\text{exchP}) \Rightarrow$

$\text{value0}(p) = \text{value1}(p)$

$p \in \text{dom}(\text{exchV}) \Rightarrow$

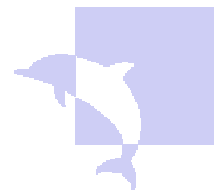
$\text{value0}(p) = \text{value1}(p) + \text{exchV}(p)$

- NB this is a forward refinement



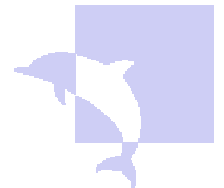
The AAAP Principle

- Keep data ***As Abstract As Possible*** when introducing algorithmic/distributed/non-atomic structure
- This will minimise proof effort when introducing algorithmic/distributed structures



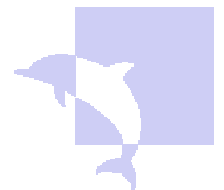
M2: transactions introduced

- Transactions
 - Are uniquely identified
 - Have attributes (from, to, amount)
 - Have abstract state: pending, ended, recoverable
 - t is pending means $\text{from}(t) \sqcap \text{to}(t) \sqcap \text{exchP}$
 - t is recoverable means
amount(t) has been added to lost(from(t))
- Actually 2 refinement models
 - M2a: remove lost but keep exchP, exchV
 - M2b: remove exchP, exchV



M3: more complex transaction state

- New operations:
 - Initialise transaction
 - StartFrom, StartTo, AbortEPA, AbortEPV etc..
- Separate states for from and to sides
 - StateF = { idleT, epr, epa, endF, abortIdleT, abortepr, abortepa }
 - StateT = { idleT, epv, endT, abortIdleT, abortepr, abortepa }
- t is recoverable when
 - from(t) is abortepa, to(t) is abortepv



Form of invariants

$\text{exchF} = \text{dom}(\text{exchP})$

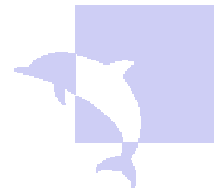
$\text{exchT} \sqsubseteq \text{exchF} \rightarrow \text{trans}$

- Alternative formulations:

$(\text{exchT} ; \text{to}) = \text{exchP}$

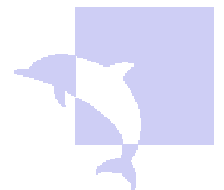
$\sqsubseteq p. (p \sqsubseteq \text{exchF} \Rightarrow$

$\text{to}(\text{exchT}(p)) = \text{exchP}(p)$



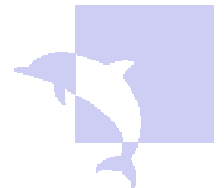
To do

- Only maintain recoverable transactions logs
- Limit the log size (lose some liveness)
- Add sequence numbers to achieve uniqueness
- Explicit messaging and message faking



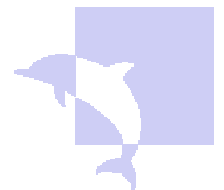
Queries about Mondex

- How is card authorisation enforced?
 - What prevents me from creating a fake card?
- How is money added to the system?
- Can a card be involved in more than one transaction?
 - Enforceable with 2-slot wallet, but not over a network
- What kind of messaging security is used?
- Multiple currencies?



Tools are Critical

- B4free not just a proof tool
 - Generates proof obligations
 - Fairly powerful automatic prover and facilities for re-running proofs on modified models
 - Guides in the construction of gluing invariants
- Removes a lot of mundane work, allowing effort to focus on the real challenges
- Evolution of refinement chain would be impractical without these tools



B4free is not perfect

- Poor factorisation of *some* POs involving nondeterministic choice (ANY x WHERE...)
- No support for fine-tuning model during interactive proof
- Some Event-B features hand-coded
- These are addressed by RODIN tools...
- <http://sourceforge.net/projects/rodin-b-sharp/>

