

Formal refinement in Z: Mondex electronic purse

Susan Stepney

Logica UK Ltd

1998

The application

NatWest Development Team

Mondex electronic purse

Smart Card for electronic commerce

Autonomous: no external control ==>
all security on card

The difficulty

ITSEC: E1 -- E6 (highest)
increasing formality

E6: *formal SP (abstract), formal
architecture (concrete),
correctness proofs*
often thought to be impossible!

Summary: the approach

Functional security properties

A correctness proof

Rigorous hand proofs

Abstract and concrete models

Not everything modelled

Security properties

1. "No value created"

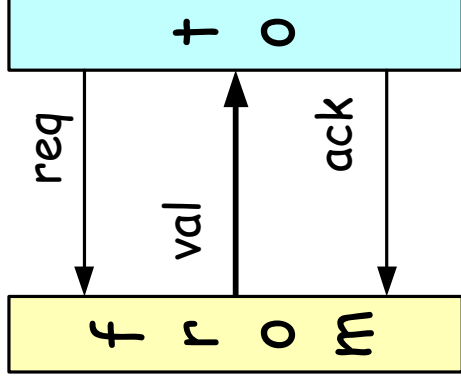
$$\Sigma \text{ £ } \geq \Sigma \text{ £ }'$$

2. "All value accounted"

3. "This transfer permitted"
(classes, etc)

Formal Z models

- Abstract model
 - promoted world of 'purses'
 - atomic value transfers
- Concrete model
 - promoted world of 'purses'
 - n-step value transfer protocol
 - logging protocol
 - ether of protocol messages



Modelled by other means

Eg cryptography: instead, ether
some messages always present \Rightarrow
forgeable
some injected only by cards \Rightarrow
protected 'somehow'
(strength of mechanism arguments)

Correctness proof

Prove that abstract Security Policy is captured by concrete architecture model

Here, SP comprises *functional* properties, which are preserved by *refinement*

Proof style

rigorous hand proof

(not deep: cut, one point, thin, Leibniz, Z
toolkit laws, ...)

structure with lemmas

tools: type-checker (fuzz/Formalizer)

human evaluators/reviewers

Summary: proof problems

Resolution of non-determinism

Back to first principles

Deriving 'backwards' rules

Finalisation / i-o refinement

Two refinements required

Resolution of non-determinism

Concrete before
Abstract (Spivey)

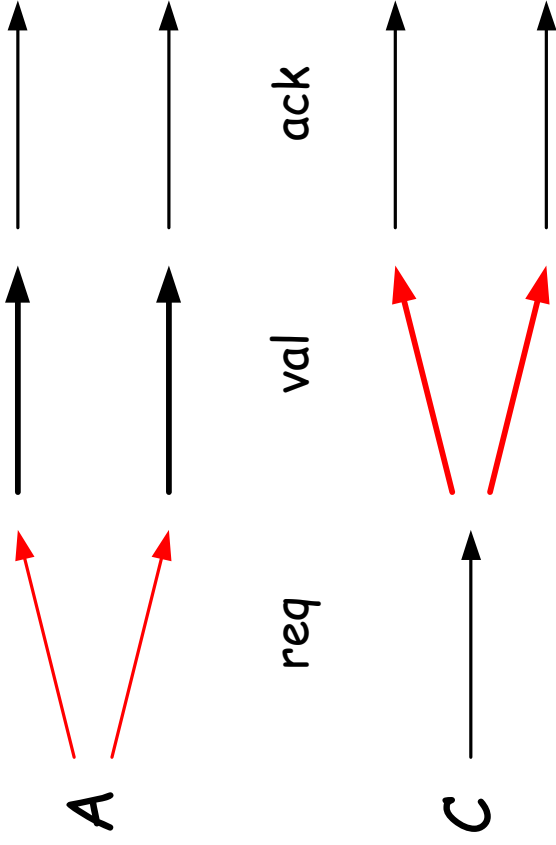
or

Abstract before
Concrete (ours)

classic Spivey

proof rules

not sufficient



Back to first principles

So, what *is* refinement?

consultancy from Jim Woodcock

He, Hoare & Sanders paper

refinement rules \Rightarrow a semantics for Z
state-and-operations specifications

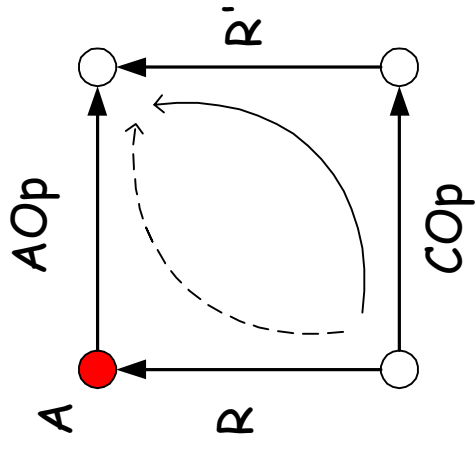
Deriving 'backwards' rules

derived from first principles

now published in
'Woodcock & Davies'

also rederived the
Spivey rules this way

as a sanity check
and confidence booster!



$COp : R'$

\vdash

$\exists A \bullet R \wedge AOp$

Finalisation

observability

(eg, *Stack* with no *Top* operation)

Not all properties of our model are
observed by i/o: so we have a non-
trivial finalisation

Input-output refinement

computational model

No good abstraction of 'balance enquiry', so:

- abstract: no i/o, + finalisation
- concrete: some i/o, + smaller finalisation

Not sufficient

Forward or backward rules alone
not sufficient to prove *all*
refinements

We needed a 2-step refinement

- 1) atomic transfer --> protocol + global constraints
- 2) global constraints --> unconstrained world

Summary: results

What was proved

Sizes and timescales

Incremental development

FM not the bottleneck

Future developments

What was proved

Proved the design:

- that the security properties hold
- that the protocol implements atomic transfer with error detection
- that local on-card constraints implement the required global constraints

Spec and proof sizes

abstract SP model: ~ 20 pages

concrete FAD model: ~ 60 pages

hand proof: ~ 200 pages

other derivations: ~ 100 pages

technical monograph PRG-126

Mondex reduced functionality
specification, and proof, 230 pages



Incremental development

2 versions: first 'reduced functionality' of Swindon pilot, then upgraded to full 'roll-out functionality'

Main change: multiple currencies

balance : \mathbb{N} became *pocket* : $CURR \leftrightarrow \mathbb{N}$

(*not* bag $CURR$, ie not *pocket* : $CURR \leftrightarrow \mathbb{N}_1$)

Not a bottleneck

Success!

Found an error in the logging protocol
FM work *ahead* of schedule,
favourable evaluation report: so,
requirement for FM is no bar to E6

The future

Was so successful --> now going for
E6 approach on other products!

Hand proof enlightening (but tedious):
looking at proof tools --- CADiZ