Reasoned Modelling

Towards Decision Support for System Designers

Andrew Ireland

Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh

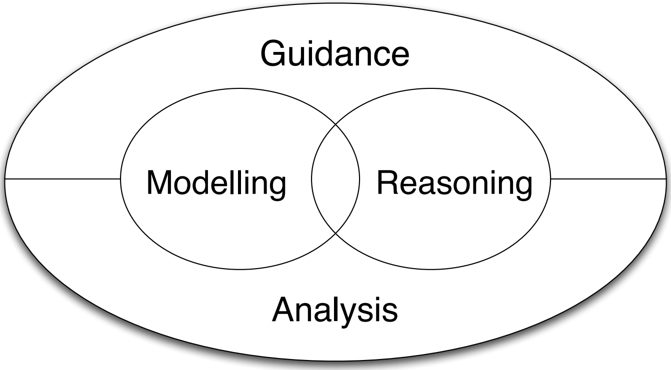
joint work with

Gudmund Grov, Maria Teresa Llano, Alison Pease

Motivations & Sponsors

- While the rigour of building formal models brings significant benefits, formal reasoning remains a major barrier to the wider acceptance of formalism within design.
- We aim to abstract away from the complexities of low-level proof obligations, providing high-level modelling guidance – we call this **reasoned modelling**.
- Accessibility and productivity – allow smart designers to make better use of their time.
- Funding:
  - REMO: EPSRC EP/F035594, EP/F037058, EP/F036647
  - AI4FM: EPSRC EP/H024204, EP/H023852, EP/H024050
  - BAE Systems (Warton)
  - EPSRC Platform Grant EP/H024204

Reasoned Modelling Vision



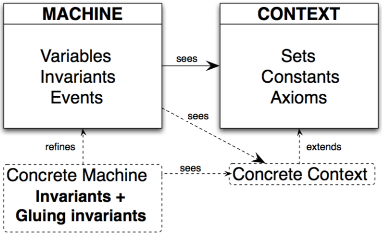
- Automated modelling guidance for designers using:
    - Common patterns of modelling and reasoning.
    - Simulation, proof-failure analysis and automated theory formation.
- Generic vision, using Event-B as our starting point.

Outline

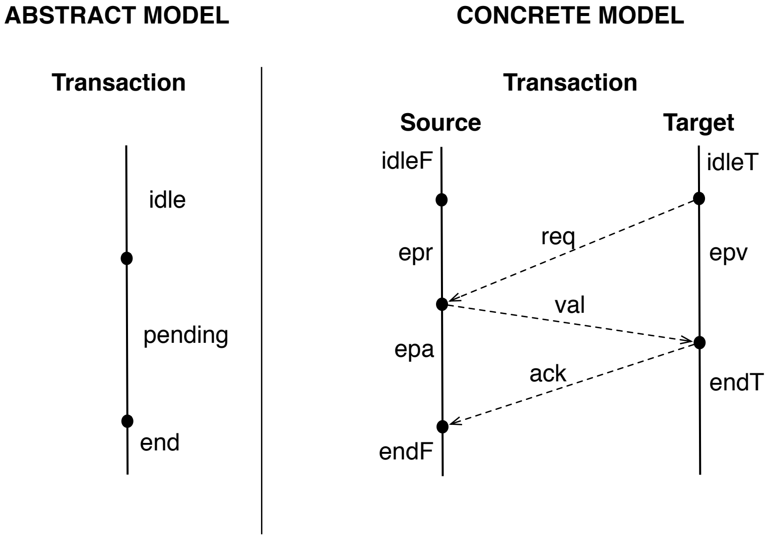
- Event-B and refinement based modelling.
- Automated theory formation in support of refinement.
- Experimental results.
- Exploiting common patterns of refinement.
- Limitations and future work.

Event-B

- A formal modelling notation and method *[Jean-Raymond Abrial]*.
- Event-B evolved from the B-Method – *focus on systems rather than sequential software.*
- Promotes an incremental style of modelling via a posit-and-prove style of refinement.
- Refinement steps are verified via mathematical proof.
- Rodin tool-set – supported via DEPLOY EU Project.

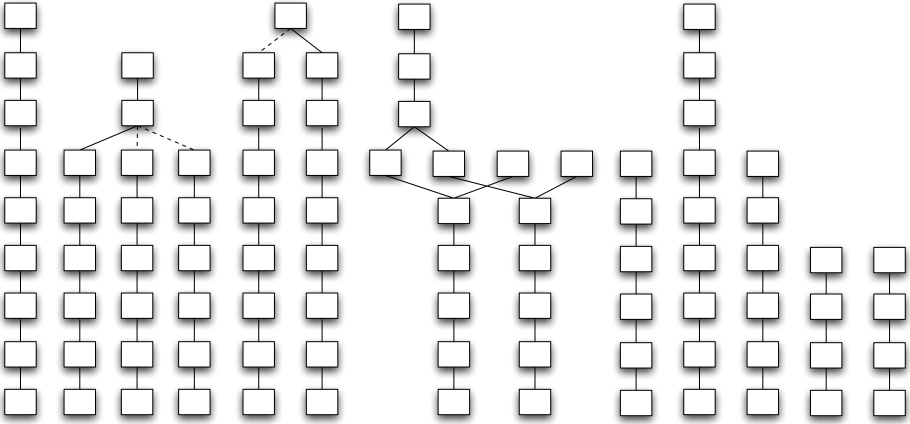


Mondex: An Electronic Payment System

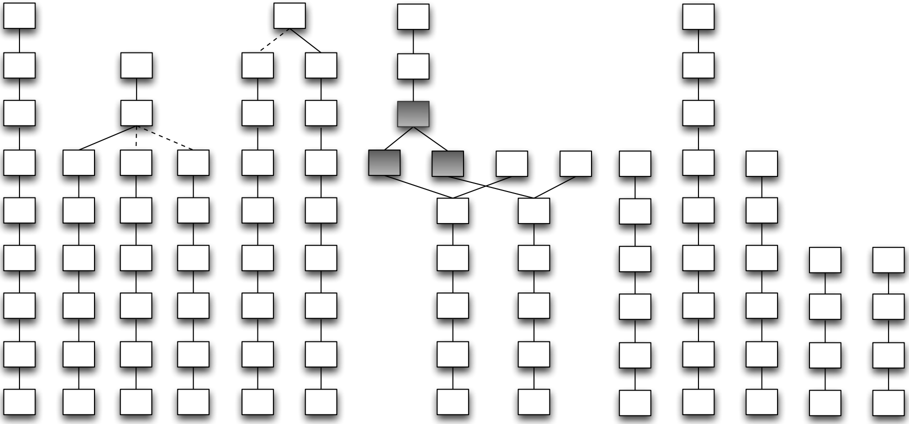


A reconstruction of “An Incremental Development of the Mondex System in Event-B”, Butler and Yadav, *Formal Aspects of Computing*, Vol. 20(1), 2008.

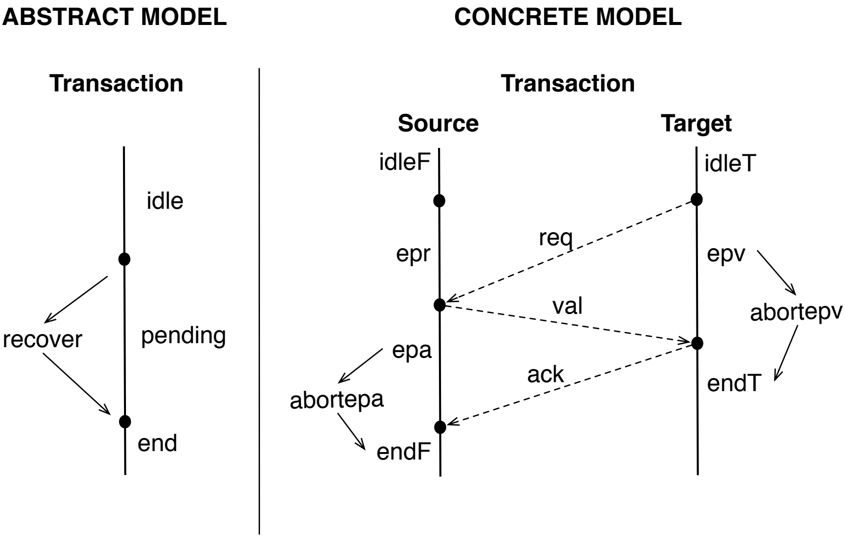
Mondex: Levels of Refinement



Mondex: Levels of Refinement



Mondex: Dealing with Transaction Failures



Mondex: Focus on a Refinement Step

Abstract

event TransferFail $\widehat{=}$
  any
    t
  where
    t $\in$ pending
  then
    recover:=recover $\cup$ {t}
    pending:=pending $\setminus$ {t}
end

Concrete

event Abortepv$_1$ $\widehat{=}$
  any
    t
  where
    t $\in$ epv
    t $\in$ abortepa
  then
    abortepv:=abortepv $\cup$ {t}
    epv:=epv $\setminus$ {t}
end

event Abortepa$_1$ $\widehat{=}$
  any
    t
  where
    t $\in$ epa
    t $\in$ abortepv
  then
    abortepa:=abortepa $\cup$ {t}
    epa:=epa $\setminus$ {t}
end

Mondex: Focus on a Refinement Step

Abstract

event TransferFail $\widehat{=}$
  any
    t
  where
    t $\in$ pending
  then
    recover:=recover $\cup$ {t}
    pending:=pending$\setminus${t}
end

Concrete

event Abortepv$_1$ $\widehat{=}$
  any
    t
  where
    t $\in$ epv
    t $\in$ abortepa
  then
    abortepv:=abortepv $\cup$ {t}
    epv:=epv$\setminus${t}
end

event Abortepa$_1$ $\widehat{=}$
  any
    t
  where
    t $\in$ epa
    t $\in$ abortepv
  then
    abortepa:=abortepa $\cup$ {t}
    epa:=epa$\setminus${t}
end

Proof Failures

GRD PO: Abortepv$_1$

t $\in$ epv
t $\in$ abortepa
$\nvdash$
t $\in$ pending

GRD PO: Abortepa$_1$

t $\in$ epa
t $\in$ abortepv
$\nvdash$
t $\in$ pending

Mondex: Focus on a Refinement Step

Abstract

event TransferFail $\widehat{=}$
  any
    t
  where
    t $\in$ pending
  then
    recover:=recover $\cup$ {t}
    pending:=pending\{t}
end

Concrete

event Abortepv$_1$ $\widehat{=}$
  any
    t
  where
    t $\in$ epv
    t $\in$ abortepa
  then
    abortepv:=abortepv $\cup$ {t}
    epv:=epv\{t}
end

event Abortepa$_1$ $\widehat{=}$
  any
    t
  where
    t $\in$ epa
    t $\in$ abortepv
  then
    abortepa:=abortepa $\cup$ {t}
    epa:=epa\{t}
end

Proof Successes

GRD PO: Abortepv$_1$

$epv \cap (epa \cup abortepa) \subseteq pending$
...
t $\in$ epv
t $\in$ abortepa
$\vdash$
t $\in$ pending

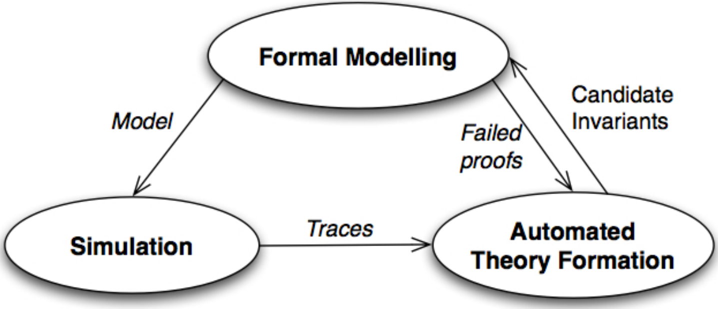
GRD PO: Abortepa$_1$

...
$epa \cap (epv \cup abortepv) \subseteq pending$
t $\in$ epa
t $\in$ abortepv
$\vdash$
t $\in$ pending

Invariants

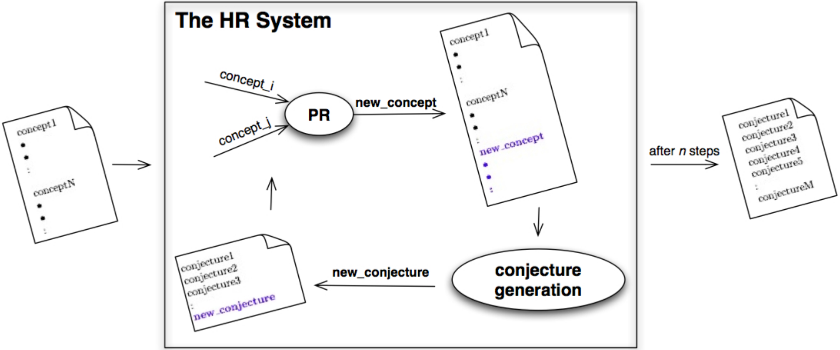
$epv \cap (epa \cup abortepa) \subseteq pending$
$epa \cap (epv \cup abortepv) \subseteq pending$

Proof-failure Driven Theory Formation



Automated Theory Formation and HR

- ATF is a machine learning technique that builds theories about objects of interest within a given domain.
- HR[1] is a system that implements ATF *[Simon Colton]*.



[1] http://www.doc.ic.ac.uk/~sgc/hr/

Concepts

- Core concepts describe the domain – represent an input to HR.
- Non-core concepts are generated iteratively by HR.
- Core concepts within Event-B denote variables, constants and sets, e.g. concepts arising from the Mondex case study include:
  - **pending**, **epv**, **epa**, ... (variables)
  - **EPV**, **EPA**, ... (constants)
  - **PURSE**, **TRANS**, ... (sets)
- Concepts are defined in terms of examples, which are represented as data tables.
- Animation traces provide a source of examples for our Event-B developments ...

Mondex: An Animation Trace

	Variables				
State	**pending**	**epa**	**epv**	**abortepa**	**abortepv**
s00	-	-	-	-	-
s01	-	-	-	-	-
:	:	:	:	:	:
s20	t5	t5	t5,t9	-	-
s21	t5	t5	t5,t9	-	-
s22	t5	t5	t5,t9	-	-
s33	t9	t9	t4,t7,t9	-	t6
s34	t9	t9	t4,t7	-	t6, t9
s35	t9	t9	t3,t4,t7	-	t6, t9
s36	t9	t9	t3,t4	-	t6, t7, t9
s47	t10	t10	t10	-	t1, t3, t4, t6, t7
s48	t10	t10	t10	-	t1, t3, t4, t6, t7
s49	t10	-	t10	t10	t1, t3, t4, t6, t7
s50	t10	-	t10	t10	t1, t3, t4, t6, t7
s51	t10	-	t8, t10	t10	t1, t3, t4, t6, t7
s52	t3,t10	t3	t8, t10	t10	t1, t3, t4, t6, t7
:	:	:	:	:	:

Mondex: Data Tables for Core Concepts

For each state, data tables are generated from the animation trace for the core concepts i.e. **pending**, **epa**, **epv**, **abortepa** and **abortepv**.

state(A)
s00
s01
:
s20
s21
s22
s33
s34
s35
s36
s47
s48
s49
s50
s51
s52
:

pending(A,B)	
s00	-
s01	-
:	:
s20	t5
s21	t5
s22	t5
s33	t9
s34	t9
s35	t9
s36	t9
s47	t10
s48	t10
s49	t10
s50	t10
s51	t10
s52	t3
s52	t10
:	:

epa(A,B)	
s00	-
s01	-
:	:
s20	t5
s21	t5
s22	t5
s33	t9
s34	t9
s35	t9
s36	t9
s47	t10
s48	t10
s49	-
s50	-
s51	-
s52	t3
:	:

abortepa(A,B)	
s00	-
s01	-
:	:
s20	-
s21	-
s22	-
s33	-
s34	-
s35	-
s36	-
s47	-
s48	-
s49	t10
s50	t10
s51	t10
s52	t10
:	:

epv(A,B)	
s00	-
s01	-
:	:
s20	t5
s20	t9
s21	t5
s21	t9
s22	t5
s22	t9
s33	t4
s33	t7
s33	t9
s34	t4
s34	t7
s35	t3
s35	t4
s35	t7
s36	t3
s36	t4
s47	t10
s48	t10
s49	t10
s50	t10
s51	t8
s51	t10
s52	t8
s52	t10
:	:

abortepv(A,B)	
s00	-
s01	-
:	:
s20	-
s21	-
s22	-
s33	t6
s34	t6
s34	t9
s35	t6
s35	t9
s36	t6
s36	t7
s36	t9
s47	t1
s47	t3
s47	t4
s47	t6
s47	t7
s48	t1
s48	t3
s48	t4
s48	t6
s48	t7
s49	t1
s49	t3
s49	t4
:	:

Production Rules & Conjecture Generation

- Production rules are used to form new non-core concepts, e.g.

  Compose: conjunction of two existing concepts.
  Disjunct: disjunction of two existing concepts.
  Negate: constructs the complement of an existing concept.
  Split: restricts an existing concept.

- New concepts trigger the search for conjectures, i.e.

  Equivalence: concepts that share that same data tables
  Implications: a data table is subsumed by another data table
  Non-existence: conjectures - an empty data table

Mondex: Disjunct Production Rule

epa(A,B)	
s00	-
s01	-
⋮	⋮
s20	t5
s21	t5
s22	t5
s33	t9
s34	t9
s35	t9
s36	t9
s47	t10
s48	t10
s49	-
s50	-
s51	-
s52	t3
⋮	⋮

abortepa(A,B)	
s00	-
s01	-
⋮	⋮
s20	-
s21	-
s22	-
s33	-
s34	-
s35	-
s36	-
s47	-
s48	-
s49	t10
s50	t10
s51	t10
s52	t10
⋮	⋮

$\xrightarrow{\textit{disjunct}<>}$

epa ∨ abortepa	
s00	-
s01	-
⋮	⋮
s20	t5
s21	t5
s22	t5
s33	t9
s34	t9
s35	t9
s36	t9
s47	t10
s48	t10
s49	t10
s50	t10
s51	t10
s52	t3
s52	t10
⋮	

Mondex: Compose Production Rule

epv(A,B)	
s00	-
s01	-
:	:
s20	t5
s20	t9
s21	t5
s21	t9
s22	t5
s22	t9
s33	t4
s33	t7
s33	t9
s34	t4
s34	t7
s35	t3
s35	t4
s35	t7
s36	t3
s36	t4
s47	t10
s48	t10
s49	t10
s50	t10
s51	t8
s51	t10
s52	t8
s52	t10
:	:

epa ∨ abortepa	
s00	-
s01	-
:	:
s20	t5
s21	t5
s22	t5
s33	t9
s34	t9
s35	t9
s36	t9
s47	t10
s48	t10
s49	t10
s50	t10
s51	t10
s52	t3
s52	t10
:	:

$\xrightarrow{\mathit{compose}<1,2>}$

epv ∧ (epa ∨ abortepa)	
:	:
s20	t5
s21	t5
s22	t5
s33	t9
s47	t10
s48	t10
s49	t10
s50	t10
s51	t10
s52	t10
:	:

Mondex: Implication Conjecture

epv ∧ (epa ∨ abortepa)	
:	:
s20	t5
s21	t5
s22	t5
s33	t9
s47	t10
s48	t10
s49	t10
s50	t10
s51	t10
s52	t10
:	:

$\Rightarrow$

pending(A,B)	
s00	-
s01	-
:	:
s20	t5
s21	t5
s22	t5
s33	t9
s34	t9
s35	t9
s36	t9
s47	t10
s48	t10
s49	t10
s50	t10
s51	t10
s52	t3
s52	t10
:	

$\forall$ A,B . state(A) $\wedge$ trans(A,B) $\wedge$ epv(A,B) $\wedge$ (epa(A,B) $\vee$ abortepa(A,B)) $\Rightarrow$ pending(A,B)

or,

Invariant: epv $\cap$ (epa $\cup$ abortepa) $\subseteq$ pending

Problems and Solutions

- HR is general-purpose – for the given Mondex refinement step, it generates **13697** conjectures in 25 seconds (1000 theory formation steps).
- HR must be tailored to Event-B style modelling, and the specifics of each refinement step.
- We use automatic proof-failure analysis coupled with heuristics, i.e.
    - Configuration Heuristics (CH): tailor production rule set.
    - Selection Heuristics (SH): focus on "interesting" conjectures.

Configuration Heuristics for HR

GRD PO: $\mathsf{Abortepv}_1$:
$t \in epv$
$t \in abortepa$
$\nvdash$
$t \in pending$

GRD PO: $\mathsf{Abortepa}_1$:
$t \in epa$
$t \in abortepv$
$\nvdash$
$t \in pending$

CH1. *Prioritise core and non-core concepts that occur within the failed POs*, e.g.

pending, epv, abortepa, epa, abortepv, *others...*

CH2. *Select production rules which will give rise to conjectures relating to the concepts occurring within the failed POs*, e.g. models that are represented using sets,

compose, disjunct and negate production rules.

CH3. *Generate conjectures that fit the shape of the expected invariants*, e.g.

implication conjectures

HR generates a total of 7036 conjectures.

Selection Heuristics for HR Generated Conjectures

SH1. *Select conjectures that focus purely on prioritised core and non-core concepts.*, e.g. 30.

SH2. *Select conjectures where the sets of variables occurring on the left- and right-hand sides are disjoint*, e.g. 30.

SH3. *Select only the more general conjectures*, e.g. 9.

SH4. *Select conjectures that discharge the failed POs*, e.g. 5.

SH5. *Select conjectures that minimise the number of additional proof failures that are introduced*, e.g. 2.

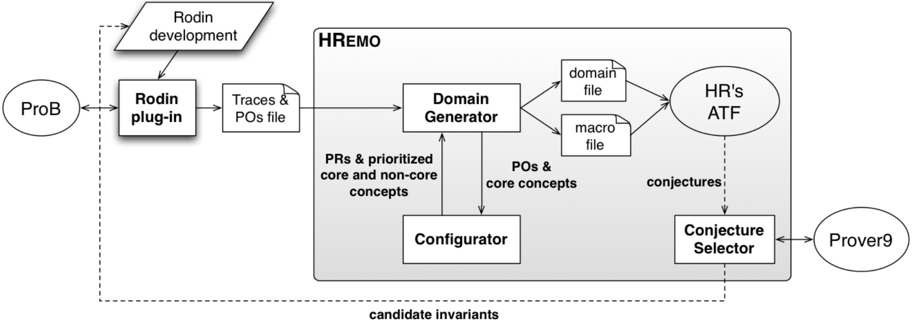
$$\forall\ A,B\ .\ \mathrm{state}(A) \wedge \mathrm{trans}(A,B) \wedge \mathrm{epv}(A,B) \wedge (\mathrm{epa}(A,B) \vee \mathrm{abortepa}(A,B)) \Rightarrow \mathrm{pending}(A,B)$$
$$\forall\ A,B\ .\ \mathrm{state}(A) \wedge \mathrm{trans}(A,B) \wedge \mathrm{epa}(A,B) \wedge (\mathrm{epv}(A,B) \vee \mathrm{abortepv}(A,B)) \Rightarrow \mathrm{pending}(A,B)$$

$$\equiv$$

Invariant 1: epv $\cap$ (epa $\cup$ abortepa) $\subseteq$ pending
Invariant 2: epa $\cap$ (epv $\cup$ abortepv) $\subseteq$ pending

Note: the above results were obtained by focusing on core concepts that occur within the PO goals, i.e. **pending**.

Experimental Framework



Experimental Results

Event-B model	Step	Failed POs	Invariants			
			Automatically discovered			
			Glue	System	Total	Iteration
Traffic light	Level 1-2	2	2	0	**2**	1
Vending machine (arith)	Level 1-2	6	3	0	**3**	1
Vending machine (sets)	Level 1-2	6	3	0	**3**	1
Cars on a bridge	Level 1-2	2	1	0	**1**	1
	Level 2-3	6	0	5	**5**	1
	Level 3-4	7	0	5	**5**	1
Mondex	Level 3-4	5	4	0	**4**	1
		6	1	4	**5**	2
		1	0	1	**1**	3
	Level 4-5	3	0	3	**3**	1
		5	0	4	**4**	2
		4	0	2	**2**	3
	Level 8-9	14	10	0	**10**	1

Automatically Discovered Invariants

Case study	Step	Automatically discovered Invariants		
		Iteration 1	...	...
Traffic Light	Level 1-2	r_light=TRUE $\vee$ a_light=TRUE $\Leftrightarrow$ red_light=TRUE sold $=$ soldMilk $+$ soldPlain		
Vending machine (arith rep.)	Level 1-2	stock $=$ stockMilk $+$ stockPlain g_light $=$ TRUE $\Leftrightarrow$ green_light $=$ TRUE givenCoin $=$ EMPTY_COIN $\Leftrightarrow$ coin $=$ NO_COIN		
Vending machine (sets rep.)	Level 1-2	available $=$ productStatus^{-1}[{AVAILABLE}] limited $=$ productStatus^{-1}[{LIMITED}] soldOut $=$ productStatus^{-1}[{SOLDOUT}]		
Cars on a bridge	Level 1-2	n $=$ a $+$ b $+$ c		
	Level 2-3	ml_tl $=$ green $\Rightarrow$ c $=$ 0 il_tl $=$ green $\Rightarrow$ a $=$ 0 ml_tl $=$ red $\Rightarrow$ ml_pass $=$ 1 il_tl $=$ red $\Rightarrow$ il_pass $=$ 1 il_tl $=$ green $\Rightarrow$ ml_tl $=$ red		
	Level 3-4	ml_out_10 $=$ TRUE $\Rightarrow$ ml_tl $=$ green il_out_10 $=$ TRUE $\Rightarrow$ il_tl $=$ green IL_IN_SR $=$ on $\Rightarrow$ A $>$ 0 IL_OUT_SR $=$ on $\Rightarrow$ B $>$ 0 ML_IN_SR $=$ on $\Rightarrow$ C $>$ 0		

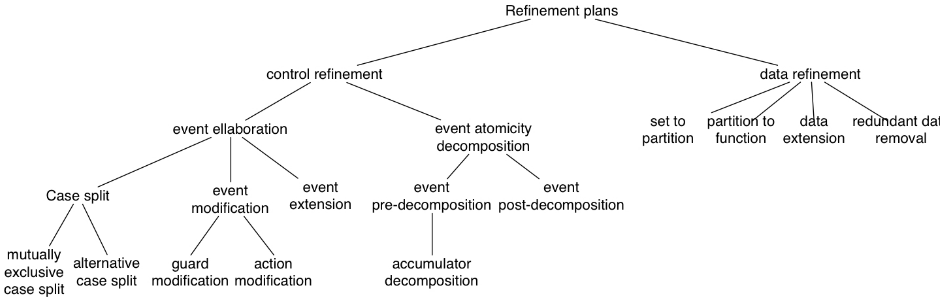
Automatically Discovered Invariants

Case study	Step	Automatically discovered Invariants		
		Iteration 1	Iteration 2	Iteration 3
Mondex	3-4	epr $\subseteq$ idle (idleF $\cup$ epr) $\subseteq$ idle epv $\cap$ (epa $\cup$ abortepa) $\subseteq$ pending epa $\cap$ (epv $\cup$ abortepv) $\subseteq$ pending abortepa $\cap$ abortepv $=$ recover	idleF $\subseteq$ idle idleT $\cap$ (epa $\cup$ abortepa) $= \varnothing$ idleT $\cap$ (epv $\cup$ abortepv) $= \varnothing$ epv $\cap$ abortepv $= \varnothing$ epa $\cap$ abortepa $= \varnothing$	epr $\cap$ idleF $= \varnothing$
	4-5	epr $\cap$ reqM $\subseteq$ epv $\cup$ abortepv epv $\cap$ valM $\subseteq$ epa $\cup$ abortepa endT $=$ endF $\cup$ ackM	reqM $\cap$ idleF $\subseteq$ epv $\cup$ abortepv valM $\cap$ idleT $= \varnothing$ epr $\cap$ valM $= \varnothing$ epr $\cap$ abortepa $= \varnothing$	valM $\cap$ idleF $= \varnothing$ abortepa $\cap$ idleF $= \varnothing$
	8-9	idleFP $=$ statusF^{-1}[{IDLEF}] eprP $=$ statusF^{-1}[{EPR}] epaP $=$ statusF^{-1}[{EPA}] aborteprP $=$ statusF^{-1}[{ABORTEPR}] abortepaP $=$ statusF^{-1}[{ABORTEPA}] endFP $=$ statusF^{-1}[{ENDF}] idleTP $=$ statusF^{-1}[{IDLET}] epvP $=$ statusF^{-1}[{EPV}] abortepvP $=$ statusF^{-1}[{ABORTEPV}] endTP $=$ statusF^{-1}[{ENDT}]		

Refinement Plans

- Combine common patterns of modelling with proof – *for now we focus on POs rather than proofs.*
- Specifically, use patterns and proof-failure analysis to tailor modelling guidance, e.g.
  - reformulation of events, invariants and axioms.
  - suggest intermediate layers of abstraction.
- Exploit synergies between automated theory formation (bottom-up) and common patterns of refinement (top-down).

Classification of Refinement Steps



event Abstract $\widehat{=}$
  any
    ...
  where
    ...
  then
    $X := Y$
    ...
end

Variables:

...

Invariants:

...

Variables:

...

Invariants:

$\neg G_1 \Rightarrow \mathcal{F}(Y, Z)$

...

event Concrete_1 $\widehat{=}$
  any
    ...
  where
    G_1
  then
    $Z := Z_0$
    ...
end

event Concrete_2 $\widehat{=}$
  any
    ...
  where
    G_2
  then
    $Z := Z \bigoplus W$
    ...
end

event Concrete_3 $\widehat{=}$
  any
    ...
  where
    G_3
  then
    $X := Z$
    ...
end

Refinement Plan: Partition-to-Function

Sets:
S
$\ldots$
Constants:
$\ldots$
Axioms:
$\ldots$

Variables:
$S_1, \ldots, S_n$
$\ldots$
Invariants:
$partition(S, S_1, \ldots, S_n)$
$\ldots$

event Abstract $\widehat{=}$
any
X
where
$X \in S_i$
then
$S_i := S_i \setminus \{X\}$
$S_j := S_j \cup \{X\}$
end

Sets:
T
$\ldots$
Constants:
$T_1, \ldots, T_n$
$\ldots$
Axioms:
$partition(T, T_1, \ldots, T_n)$
$\ldots$

Variables:
$\ldots$
Invariants:
$F \in S \to T$
$S_i = F^{-1}[\{T_m\}]$
$S_j = F^{-1}[\{T_n\}]$
$\ldots$

event Concrete $\widehat{=}$
any
X
where
$F(X) \in T_m$
then
$F(X) := T_n$
$\ldots$
end

		Control refinement				Data refinement				
Model	Refinement	Cases	Atomicity Dec. Acum	Atomicity Dec. Pre-Dec.	Event Mod.	Part2Fun	Set2Part	Data Extension	Redudant Data Removal	HR
Cars on a bridge	Ref 1						✔			✔
	Ref 2	✔✔	✔✔							✔
	Ref 3			✔						✔
Mondex	Ref 1			✔						
	Ref 2								✔	
	Ref 3	✔		✔✔						✔
	Ref 4			✔						✔
	Ref 5				✔					
	Ref 6							✔		
	Ref 7								✔	
	Ref 8					✔				✔
Flash file system	Ref 1						✔			
	Ref 2				✔					
	Ref 3				✔					
	Ref 4				✔					
	Ref 5		✔							
	Ref 6		✔							
Location access controller	Ref 1				✔					
	Ref 2			✔						
	Ref 3						✔			
	Ref 4						✔✔			
plc	Ref 1									
	Ref 2			✔						
	Ref 3						✔			
	Ref 4						✔			
	Ref 5					✔				
Network topology discovery	Ref 1									
	Ref 2			✔						
	Ref 3							✔		
	Ref 4									
	Ref 5								✔	
	Ref 6				✔					
	Ref 7							✔	✔	

Summary, Limitations & Future Work

- Reasoned modelling aims to provide decision support at the level of modelling – *freeing users from the burden of low-level proof-failures.*
- Focusing on Event-B, we have reported on complementary aspects of this programme of work:
  - An integrated approach to invariant discovery via animation, proof-failure analysis and automated theory formation.
  - Initial experiments have shown that the performance of the integrated approach can be significantly improved by exploiting common patterns of refinement.
- Performance of ATF is linked to the quality of the animation traces – work is needed on improving these traces.
- Understanding the needs of non-expert users requires engagement with "live" modelling efforts.
- Bridging the gap between formalisms such as Event-B and more conventional notations, e.g. build upon UML-B.