University of Southampton

University of Southampton

Developing the UML-B modelling tools

C. Snook,
M. Butler, T.S. Hoang, D. Dghaym, and A. Salehi Fathabadi

University of Southampton, U.K.

F-IDE 2022 Workshop
26/09/2022, Berlin, Germany

Outline

University of Southampton

- Motivation
  - Empirical assessments of formal methods
  - Cognitive Dimensions of Notations
  - Why we chose UML and B/Event-B
  - Concept – UML-B Class diagrams & State-machines
- History of development
  - V1 – Extending standard UML
  - V2 – UML-B : Like UML but different
  - V3 – iUML-B : Extending Event-B
  - V4 – xUML-B : A human usable text persistence
- Conclusions

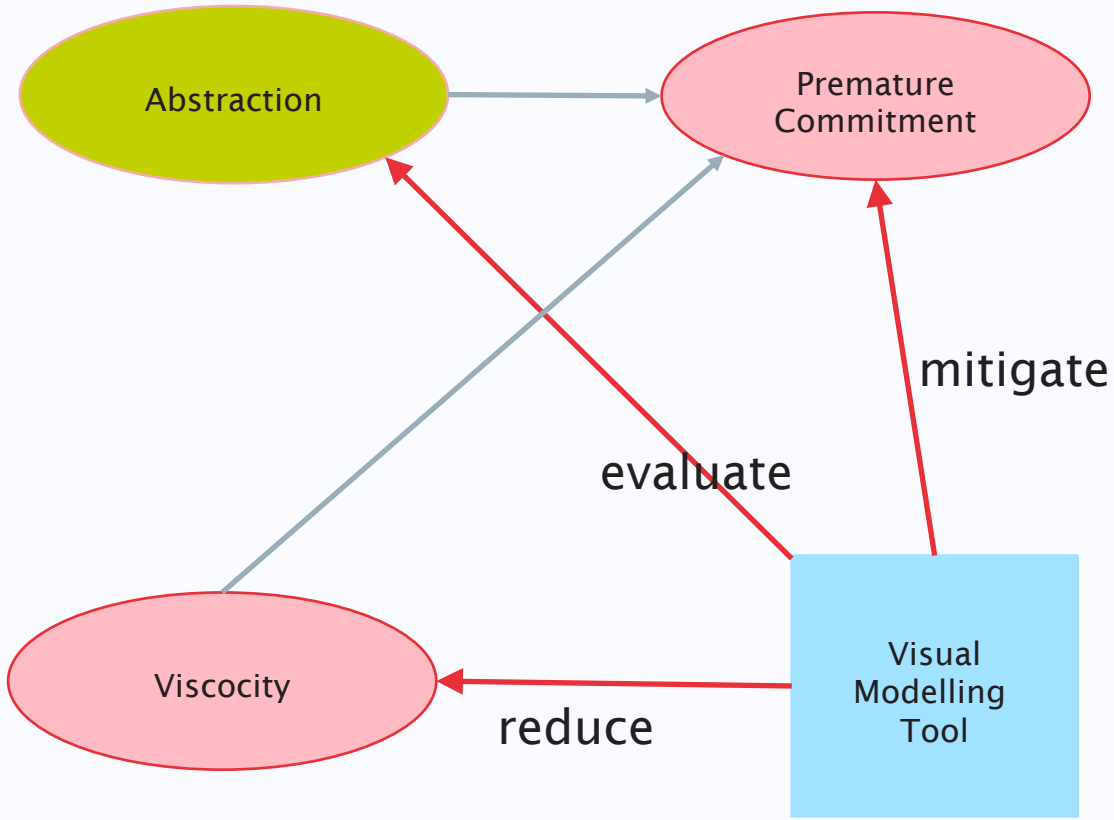
Empirical assessments (before UML-B)

- PhD 1998-2001 – “Exploring the Barriers to Formal Specification”

- Experiments on Understandability
  - Formal specifications are no more difficult to understand than programs (Z v Java)
- Surveys of FM in industry
  - “Finding useful abstractions is difficult”. J.Wordsworth, IBM

Colin Snook, Rachel Harrison
Practitioners' views on the use of formal methods: an industrial survey by structured interview, *Information and Software Technology*, 43, (4)

Why does a diagram editor help?



Cognitive Dimensions of Notations
(Blackwell and Green)

"provide a common vocabulary for discussing many factors in notation, UI or programming language design"

Rozilawati Razali, Colin Snook, Mike Poppleton & Paul Garratt (2008) Usability Assessment of a UML-based Formal Modelling Method Using Cognitive Dimensions Framework *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*

Motivation

- An approachable interface for newcomers to formal modelling
- Provide diagrams to
  - help visualise models
  - make it easier to create models
  - help explore abstractions
- Provide extra features to Event-B
  - ‘Lifting’ - instances
  - Sequencing of events
- N.b. not trying to formalise UML

Empirical Assessments (assessing UML-B)

- 2 Experiments on comprehensibility of UML-B
  - UML-B helped students comprehend models
    - 1) v B
    - 2) v Event-B
  - "The results suggest that the integration of semi-formal and formal notations expedites the subjects' comprehension tasks with accuracy even with limited hours of training"

Rozilawati Razali, Colin Snook & Mike Poppleton, (2007)
Comprehensibility of UML-based Formal Model – A Series of Controlled Experiments
At *1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies (WEASELTech) 2007*.

Why B.. and later Event-B

- B - modelling computer programs via refinement
- Event-B – modelling Systems via refinement

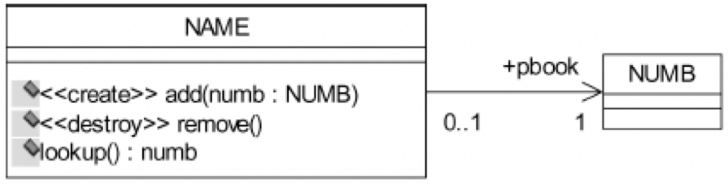
- Practical, useable
  - industrial users
- Abstraction + Refinement
  - verification by theorem provers
- Good tool support
  - B-Core/AtelierB Rodin modelling platform

Why UML

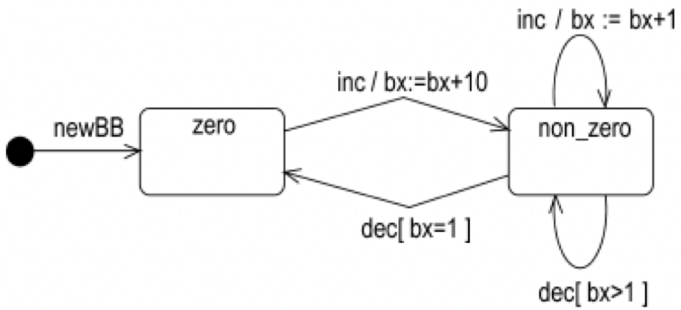
- Popular in industry
  - Trying to break down the barrier
- Class and associations → entity relationship diagrams
  - Represent sets of instances and relations (B/Event-B data)
- Statemachines
  - Transitions can represent B/Event-B operations/events

History of UML-B - Version 1

- B-UML/U2B 2000-2003 (Matisse, Pussee)
  - Based on Rational Rose UML tool
    - UML Profile (Stereotypes), textual annotations
    - Much of UML not used (i.e. not useful)
    - Some properties added to UML
      - E.g. notion of refinment
    - Some things adapted (by bending the UML semantics)
  - Generated Classical B text file to be imported into B-Core
  - Very poor integration between modelling and verification
    - Windows -> Linux

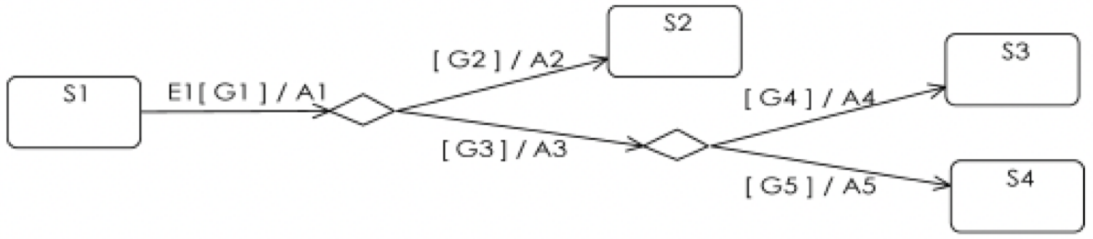


operation	guard	action
add	numb∉ran($pbook)	pbook:=numb
remove	-	-
lookup	-	numb:=pbook



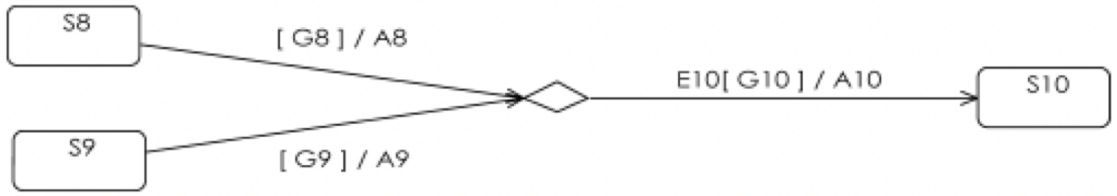
History of UML-B - Version 1 (cont.)

- B had a kind of conditional execution
  - SELECT
- Which we used in state machines
  - Decision pseudo-states (salmiakki)




```
E1 =   SELECT state= S1 ∧ G1 THEN A1 | |
       SELECT G2 THEN A2 | | state := S2
       WHEN G3 THEN A3 | |
          SELECT G4 THEN A4 | | state := S3
          WHEN G5 THEN A5 | | state := S4
          END
       END
END
```

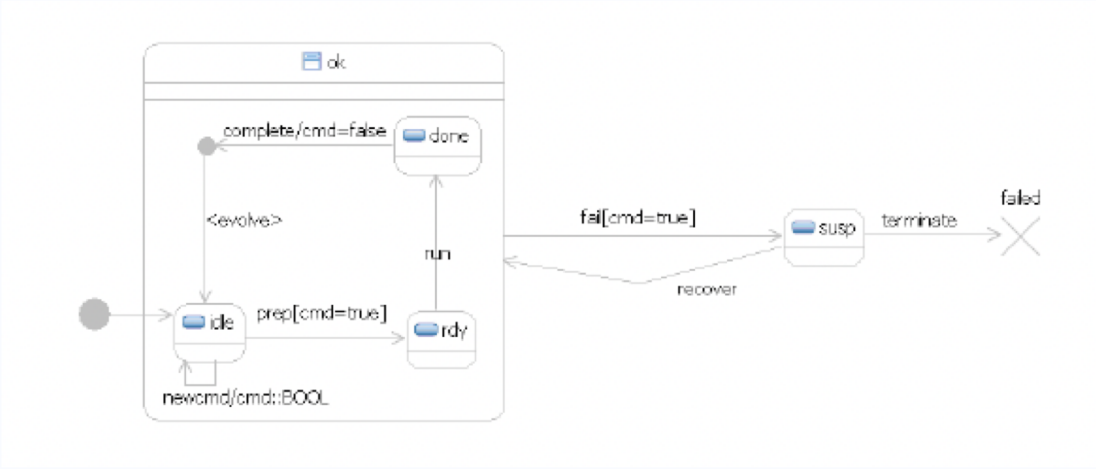
- This was lost from later versions because Event-B does not have SELECT




```
E10 = SELECT G10 THEN A10 | | state:=S10 | |
      SELECT state=S8 ∧ G8 THEN A8
      WHEN  state=S9 ∧ G9 THEN  A9 END
END
```

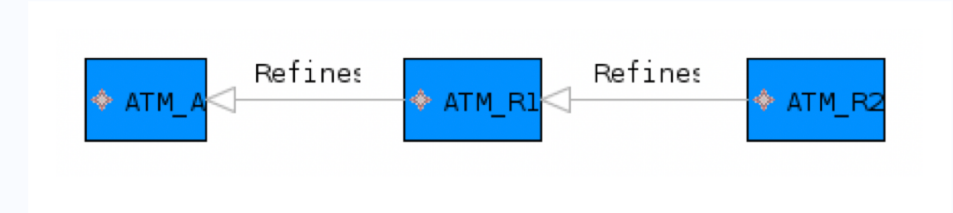
History of UML-B - Version 1.5

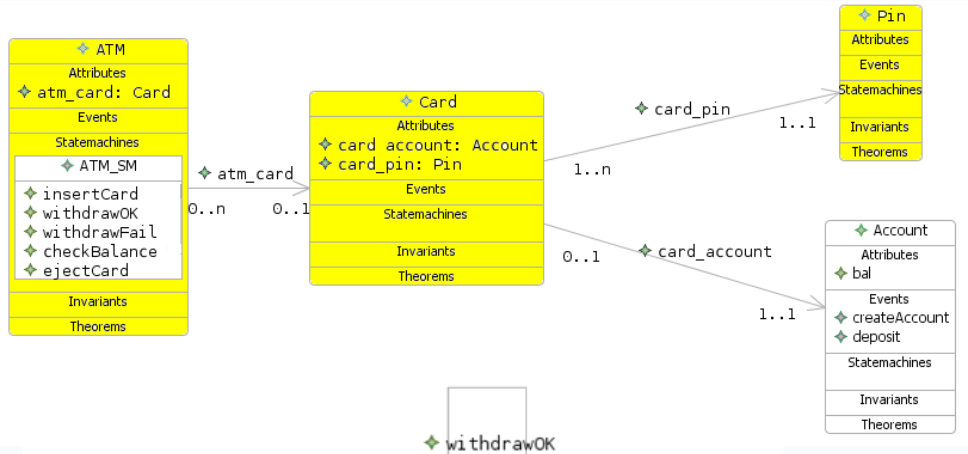
- UML-B for Event-B (first attempt) 2004-2005 (Rodin)
  - Rational Architect (Eclipse based UML tool)
  - UML Profile using the UML2 Eclipse plug-in
  - Generated Event-B for Rodin (Eclipse based)
- Much better integration than V1
  - Everything in Eclipse
- Mismatch between UML and Event-B
  - Even more than UML-B Version 1
    - since Event-B removed some of the program-like control flow (e.g. SELECT)
  - Decided to deviate from UML

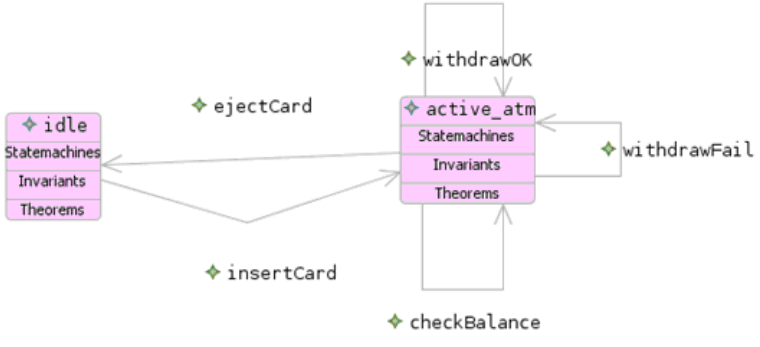


History of UML-B - Version 2

- UML-B *like UML but different* 2005-2008 (Rodin, INESS)
  - EMF meta-model for UML-B
  - UML-B model is a project
    - Generates a complete Rodin Event-B project
    - UML-B defines the machine/context structure
    - UML-B Machines - class diagrams, state-machines
  - Much cleaner language – *no more fighting UML*
  - More oriented to systems
  - But.. Forced to do everything in diagrams
    - Event-B gets overwritten
    - Event-B is Only for verification
    - Event-B users would prefer more flexible choice between text and diagram

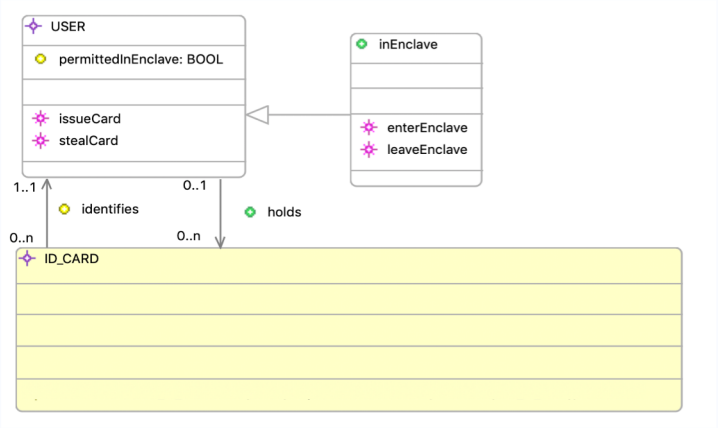


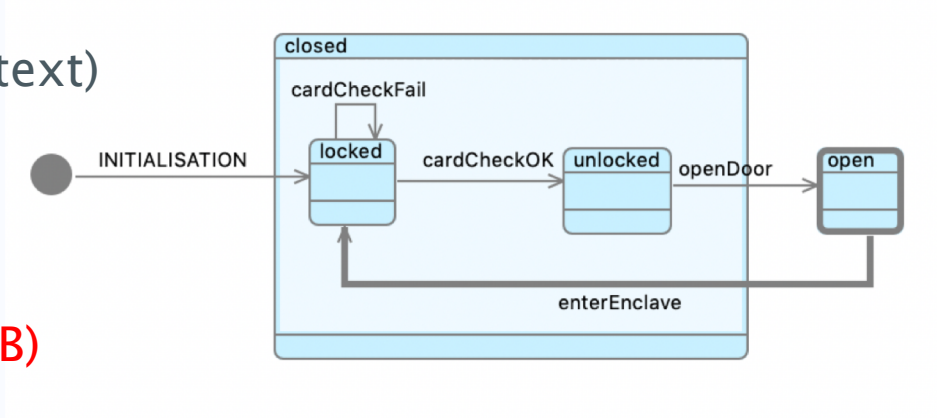


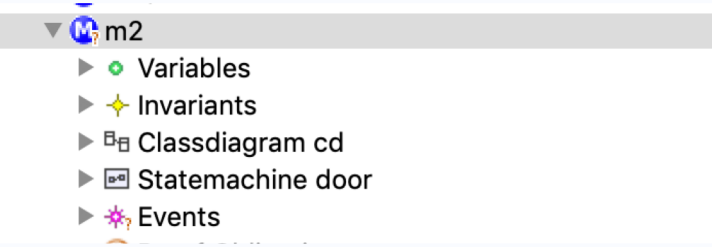


History of UML-B - Version 3

- iUML-B *integrated into Event-B* 2008 - current (Deploy, Advance)
  - EMF meta-model of Event-B
    - Has extension mechanisms
    - Extended for iUML-B
  - iUML-B model is contained in a Machine (or Context)
  - Generates Event-B elements in the parent Machine (or Context)
  - But doesn't touch any hand written Event-B
- Not compatible with CamilleX
  - CamilleX overwrites the machine (which contains the UML-B)
- Human-usable text syntax for UML-B?
  - Comparison for change tracking
  - Copy/paste







History of UML-B - Version 4

- xUML-B xtext serialisation of iUML-B 2008 - (Deploy, Advance)
  - Uses same iUML-B metamodel and diagrams but…
  - UML-B model is ***NOT*** contained in a Machine (or Context)
    - Serialised in a separate file (currently XMI)
    - *Working on* > Serialised as Xtext – Human usable text notation
      - Useful for comparing models, cut and paste etc.
  - Generates Event-B into the ***referenced*** Machine (or Context)
    - still supports hand written Event-B..
    - .. Which may be generated by CamilleX

Conclusions

University of Southampton

- Heavily featured semi-formal modelling languages such as UML are difficult to use for precise formally verified specification.
  - UML can be specialised through profiles and stereotypes,
  - but users are confused if familiar features are not used or given a different semantics.
  - Therefore, it is better not to try to translate UML but to invent a new notation that is better suited to the target formalism.

- UML-B is not UML.. Even so, users can be confused when the model does not behave as UML.
  - In UML-B, 2 Transitions with the same event but different sources are not conditional.
  - In UML statechart - ‘run to completion’ semantics.

Conclusions

- Model edition, checking and verification needs to be highly integrated so that changes can be quickly assessed.
- Some users prefer a self contained diagrammatic notation, but..
- experienced users want the flexibility to choose between diagrammatic and textual representations for different parts of a model.
- Even when diagrams are used, users express a strong desire for a human usable textual serialisation
  - helps with maintenance activities such as version comparison and copy and paste as well as enabling a quick overview of the content

Questions?

Visit UML-B.org for more info: