The Rodin Platform

Michael Butler

DEPLOY Tool-Building Team

www.deploy-project.eu

www.event-b.org

Rodin Tool for Event-B

- Extension of Eclipse IDE (Java based)
- Proof manager use a range of
- Rodin Eclipse *Builder*coordinates:
  - Well-formedness + type checker
  - Proof obligation generator
  - Proof manager
  - Propagation of changes

Rodin Proof Manager (PM)

- PM constructs proof tree for each PO
- Automatic and interactive modes
- PM manages used hypotheses
- PM calls *reasoners*to
  - discharge goal, or
  - split goal into subgoals
- Collection of reasoners:
  - simplifier, rule-based, decision procedures, …
- Basic tactics to define PM and reasoners

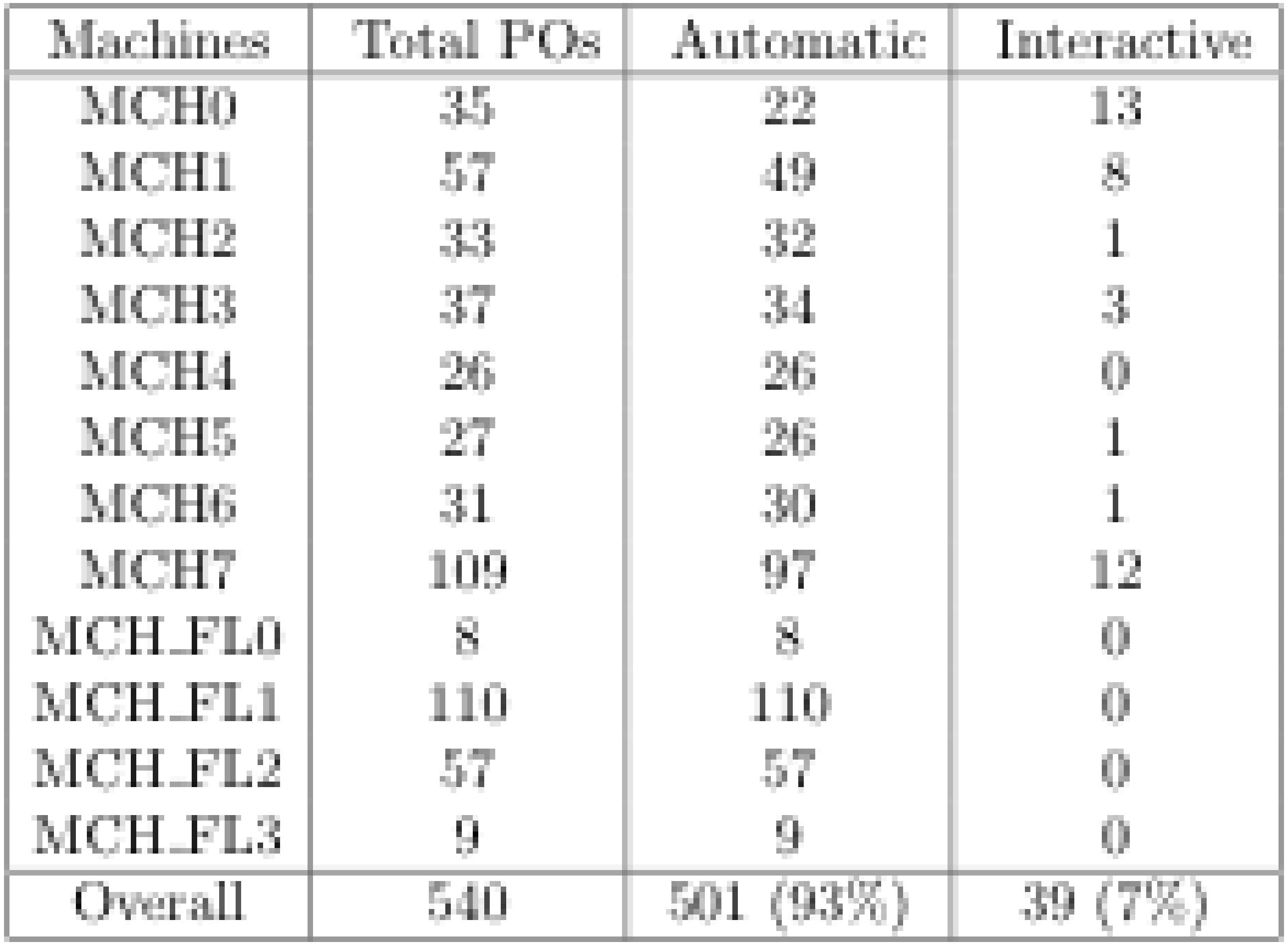
Differential proving in Rodin

- Models are constantly being changed
- When a model changes, proof impact of changes should be minimised as much as possible:

- Sufficiency comparison of POs
  - In case of success, provers return list of *used hypotheses*
  - Proof valid provided the used hypothesis are in the new version of a PO

- Simple model refactoring:
  - Identifier renaming applied to models (avoiding name clash)
  - Corresponding POs and proofs automatically renamed

Statistics from Flash-based file development in Event-B



Machines	Total POs	Automatic	Interactive
MCH0	35	22	13
MCH1	57	49	8
MCH2	33	32	1
MCH3	37	34	3
MCH4	26	26	0
MCH5	27	26	1
MCH6	31	30	1
MCH7	109	97	12
MCH_FL0	8	8	0
MCH_FL1	110	110	0
MCH_FL2	57	57	0
MCH_FL3	9	9	0
Overall	540	501 (93%)	39 (7%)

Rodin Plug-ins

- AtelierBprovers
- UML-B
- ProB: animation, consistency and refinement checking
- AnimB
- Brama
- Camille (texteditor)

Demo

On-going Developments

Improving proof automation

- Better integration with model-checking
- Extensible prover
- Extensible language
- Exploit SMT and First Order provers

Extensible language

- Rodin currently supports rich set theoretic language, but there is always a need for language extensions

- Language extended by defining Theory:
  - User-defined polymorphic operators
  - User-defined algebraic types
  - Proof rules used by rule-based prover
  - Proof rules give rise to POs

Scaling

- Team-based development
  - Parallel development: viewing conflicts / merge
  - Impact on proof (open question)

- Composition + decomposition
  - Shared variables style
  - Shared event style
  - Plug-in for decomposing models and independent refinement

Code Generation

- Introduce *algorithmic* structures
  - introduced through refinement
  - sequential and concurrent
  - data types defined in theory components
  - Back-end to Ada/C

- Event-B importer for AtelierB

Other Deploy commitments

- Requirements tracing
  - Prototype plug-in exists
  - concepts still evolving
- Reuse:
  - Instantiation of generic developments (early 2010)
  - Refinement patterns (evolving)
- Tighter integration of UML-B and Event-B
  - state machines and class diagrams within Event-B models

Wish list

- Enabledness POs
- Automatic refinement
- Support for probability
- Automated provers/SMT for set theory
  - common context
  - used hypothesis
  - extensible operator
- Reasoned modelling support
- Flexible document management

Contributors to Rodin

Jean-Raymond Abrial
Stefan Hallerstede
Farhad Mehta
Thierry Lecomte
Mathieu Clabaut
Jens Bendisposto
Dominique Cansell
Renato Silva
Carine Pascal
Michael Jastram
IssamMaamria
AbdolbaghiRezazadeh
KriangsakDamchoom
. . .

Laurent Voisin
Thai Son Hoang
Christophe Metayer
Michael Leuschel
Colin Snook
Antoine Requet
Cliff Jones
Francois Terrier
Nicolas Beauger
Fabian Fritz
Andy Edmunds
Mar Yah Said
Daniel Plagge

Keep up to date / contribute

- www.event-b.org
- wiki.event-b.org
  - – share your Event-B models
  - – share your plug-in plans
  - – suggest plug-in ideas