# Developing with the RODIN tools

**Jeremy Bryans**

21 May, 2010

# Overview

- Event-B by example
  - Models
  - Refinement
  - Proof Obligations
- Demo – The RODIN tools
- Some observations

- An EventB model contains contexts and machines
- contexts contain carrier sets, constants, axioms and theorems
- machines specify behaviour and contain variables, invariants, theorems and events

# An Event B machine

**variables:** $sent, chan, cons$

**invariants**

$sent \subseteq MSG$

$chan \subseteq MSG$

$cons \subseteq MSG$

$cons \cap chan = \varnothing$

$sent = chan \cup cons$

$cons \subseteq sent \setminus chan$

init

**begin**

$sent, chan, cons := \varnothing, \varnothing, \varnothing$

**end**

snd

**any** $m$ **where**

$m \in MSG$

$m \notin sent$

**then**

$sent := sent \cup \{m\}$

$chan := chan \cup \{m\}$

**end**

# Refinement and proof obligations

- Machines form a chain, and each machine is linked to its predecessor via a refinement relation
  - Each abstract event is refined by one (or more than one event) in the concrete machine
- Proof obligations – ensure the internal consistency of machines, and ensure that the refinement relation holds
  - Preservation of invariants – invariants continue to hold whenever variables change their values
  - Guard strengthening – concrete event is enabled no more often than it's abstract counterpart

Newcastle University

# The RODIN tool

# (Personal) observations on proving with the RODIN tools

- Sometimes the provers are better than you expect!
- Proofs of similar properties might require slightly different instantiations early on. Remainder of proof must then be replayed.
- POs resulting from updating relations can be hard work
- Appropriate case splits in proofs can take a long time (to identify, and to play through)