AI$_4$FM

AI4FM: Another way to use AI ideas to support formal methods

Cliff Jones
Newcastle University

Welcome!

AI$_4$FM

JISC list: AI4FM-INFO

AI and TP

AI$_4$FM

- one of first problems tackled
  - Herb Simon *"Logic Theory Machine"*
  - John McCarthy *"Advice Taker"*
- heuristics
  - enormously successful
  - but, always a horizon

There are many FMs

AI_4FM

- VDM
  - created in IBM Lab Vienna
- Z
- B
- Event-B
- ASMs
- …

my sort of FM

AI$_4$FM

- posit and prove
  - designer uses intuition: posits a design step
    - useful abstractions
    - documents in a formal notation
  - system generates “Proof Obligations” (POG)
  - proof discharged
    - automatically ☺
    - interactively ☹
- background
  - Rodin/DEPLOY projects
  - VxC

The problem (that we hope to ameliorate)

AI$_4$FM

- not all POs are discharged automatically
  - Steve Wright's figures
    - 5000 POs; varying stages 30%-100%
- often, when POs fail, engineers find hard
  - but POs/application follow a very similar pattern
- Abrial anecdote

The team

AI$_4$FM

- CBJ
  - plus Leo Freitas + Andrius Velykis
- Alan Bundy (Edinburgh)
  - plus Gudmund Grov + PhD
- Andrew Ireland (H-W)
- Michael Butler
- 4 years
- *this was actually intended to be Phase II*

A dichotomy

AI$_4$FM

- tackle by model revision (was to be Phase I)
  - how will model revision work with evolution?
  - time problem with large models
    - debate about avoiding “large models”
- vs
- tackle via proofs (was to be Phase II)
- so – not “vs” – we need both approaches
    - we will start with proofs

Sooo

AI$_4$FM

- we will focus on POs from FMs
  - try to be generic across FMs
- discussion with Bundy on “mining proofs”
- AI aim: learning from human TP
  - not: better (and better) heuristics
    - they always have a limit
- not even learn from finished proofs
  - learn from proof attempts
    - maybe even failures – “proof critics”

Finished proofs vs proof process

AI$_4$FM

- not just backwards (nor just forwards)
  - cf. GUI that lets user jump about
- others have analysed complete proofs
- we want to instrument tools to see how the steps evolve
- what can we learn from dead ends?