Languages and states (another view of "Why")

Cliff Jones
Newcastle University

Menu

- one technical postscript
  - ... to other talks on AI4FM project
  - recall: my *personal* focus on CxC
- a bunch of (polemic) points
  - if time permits
  - "an idealist's response to justifiable criticism"

“Models of Why”

- (as Leo said)
  - we concocted this to indicate what might be extractable from proof (attempt)s
  - of course, we have yet to perfect it!
  - nor have we published it beyond project
  - have a “state” (Σ)

Designing a language: advice

- postpone concrete syntax
  - as long as possible
- even, postpone abstract syntax
- focus on the underlying state Σ
  - (before thinking about (SOS) rules)

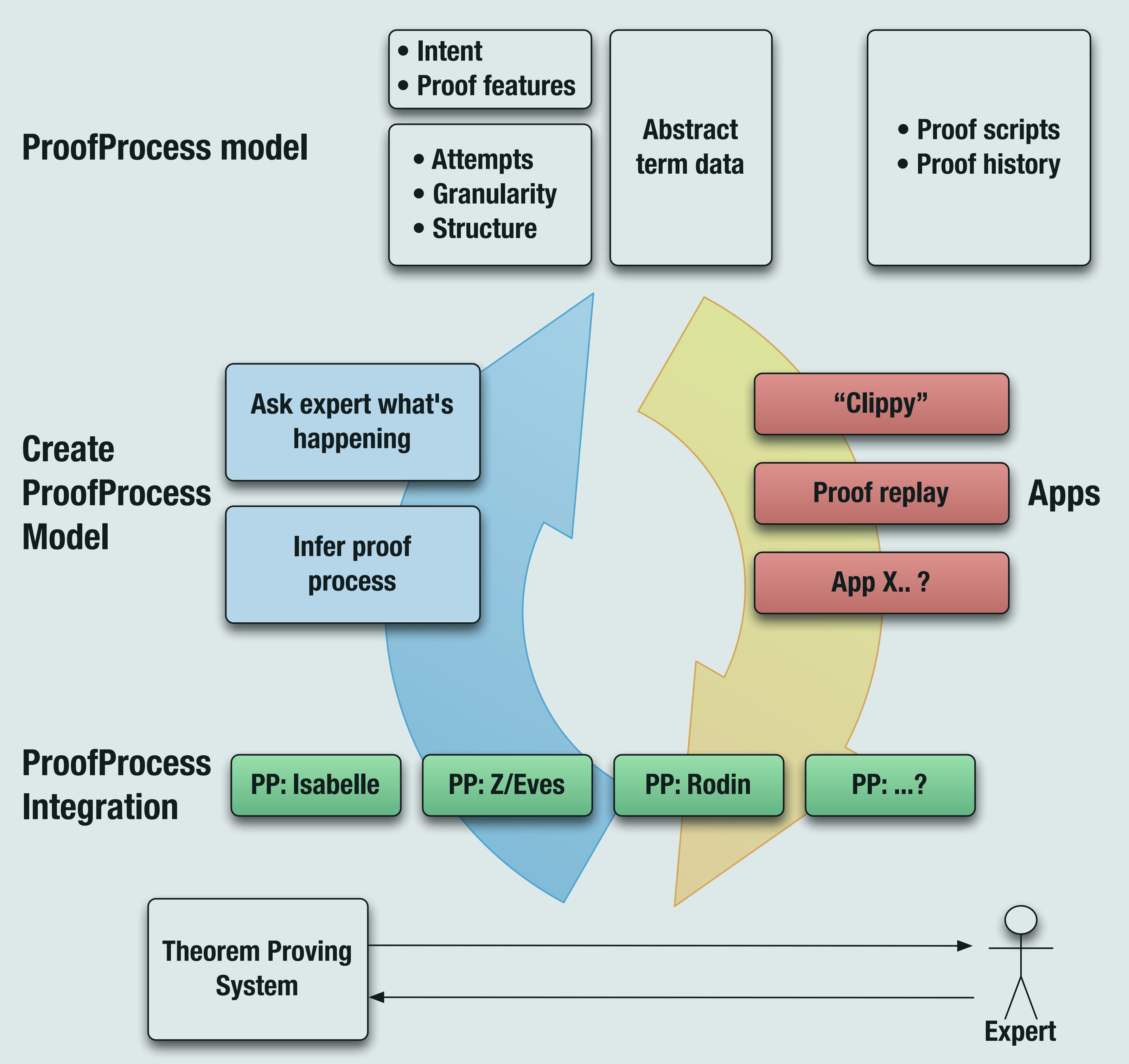
Isabelle (e.g.)

- Σ
  - pretty flat!
  - current goal list
  - ...

Change of view

- (recent)
- we are designing the state of the language that helps express learning and replay

Architecture



Models of why
as ∑

- (collection of *Theories*)
  - “inheritance” links
- defines *Operators*
- and *Conjectures*

Conjectures

- can have *Proof*
  - yes, the whole thing!
  - structured
- but can also be *Axiom, Trusted, Tool*
- (or be incomplete)
- can also have *Disproof*
  - Aaron point about negative information

Conjectures might also have

- *Shape*
- *Uses*
  - *set of Clue*

- etc., etc.

Details are unimportant today (but "vital" later)

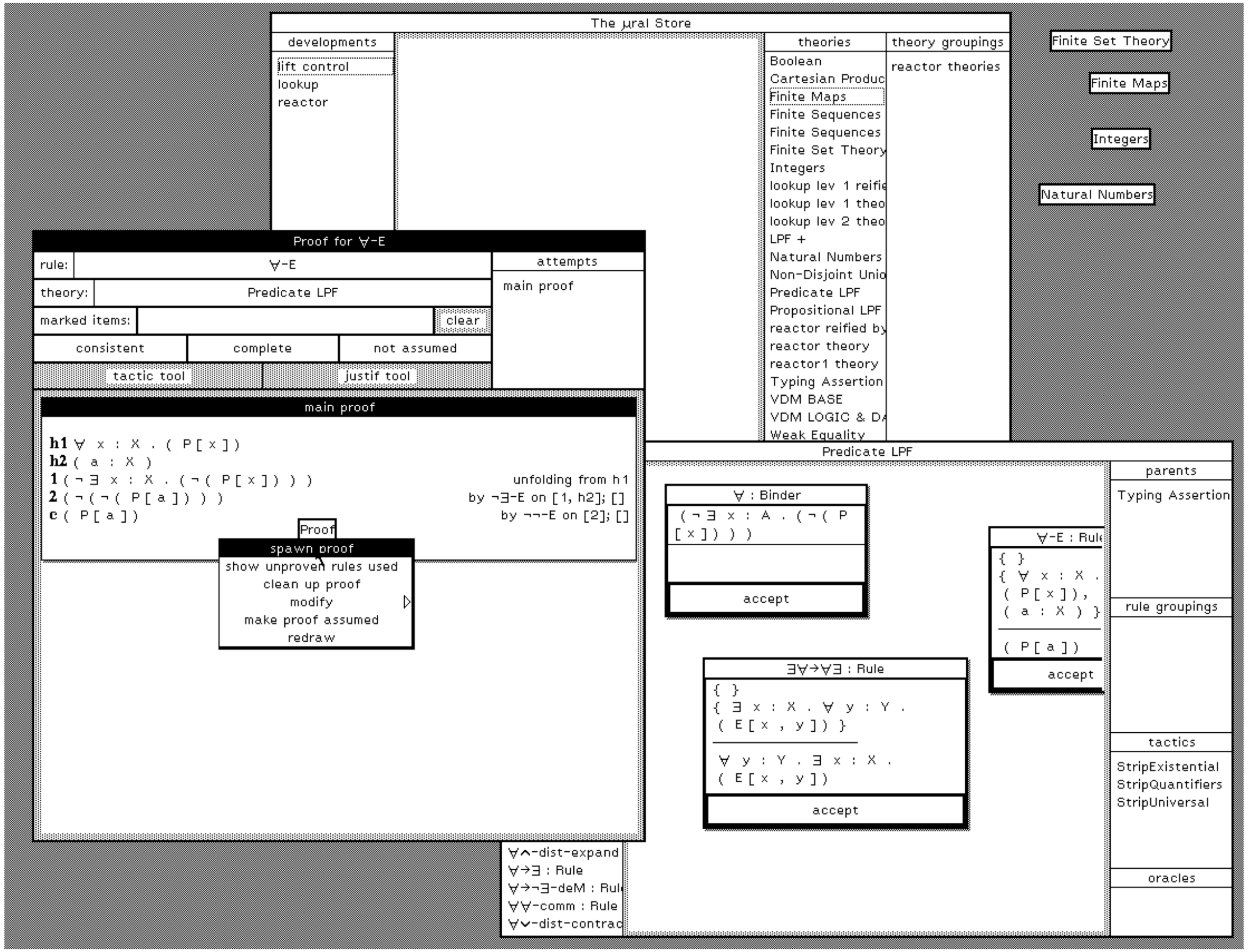
- crucial: we fix the state of this language *before* worrying about its statements
- from this viewpoint:
  - we are trying to "parse"/prompt a new attempt
  - match to previous "programs" (graph matching)
  - clear role for machine learning

Further language issues

- the language has to harness parallelism
- “non-procedural”?
- the form of this language might be unconventional
- some of activity at model (pre PO) stage

Polemics

- there will be a horizon for *any* TP ideas
- extra model layers just to reduce TP task?
- we are trying to harness an extra resource
  - the results of an expert doing one proof
- there is evidence for "families"
- lessons from "mural" (cf. Ursula's archeology)


The µral Store
developments
lift control
lookup
reactor
theories
Boolean
Cartesian Produc
Finite Maps
Finite Sequences
Finite Sequences
Finite Set Theory
Integers
lookup lev 1 reifie
lookup lev 1 theo
lookup lev 2 theo
LPF +
Natural Numbers
Non-Disjoint Unio
Predicate LPF
Propositional LPF
reactor reified by
reactor theory
reactor1 theory
Typing Assertion
VDM BASE
VDM LOGIC & D
Weak Equality
theory groupings
reactor theories
Finite Set Theory
Finite Maps
Integers
Natural Numbers
Proof for ∀-E
rule: ∀-E
theory: Predicate LPF
marked items:
clear
consistent
complete
not assumed
tactic tool
justif tool
attempts
main proof
main proof
h1 ∀ x : X . (P[x])
h2 (a : X)
1 (¬ ∃ x : X . (¬ (P[x]))) unfolding from h1
2 (¬ (¬ (P[a]))) by ¬∃-E on [1, h2]; []
c (P[a]) by ¬¬-E on [2]; []
Proof
spawn proof
show unproven rules used
clean up proof
modify
make proof assumed
redraw
Predicate LPF
∀ : Binder
(¬ ∃ x : A . (¬ (P [x])))
accept
∀-E : Rule
{ }
{ ∀ x : X . (P[x]), (a : X) }
(P[a])
accept
∃∀→∀∃ : Rule
{ }
{ ∃ x : X . ∀ y : Y . (E[x , y]) }
∀ y : Y . ∃ x : X . (E[x , y])
accept
parents
Typing Assertion
rule groupings
tactics
StripExistential
StripQuantifiers
StripUniversal
oracles
∀∧-dist-expand
∀→∃ : Rule
∀→¬∃-deM : Rul
∀∀-comm : Rule
∀∨-dist-contrac

“mural”

- interesting GUI experiment
  - but no decision procedures!
- built from formal spec
  - which was maintained!!
- [JJLM91] available as:
  - homepages.cs.ncl.ac.uk/cliff.jones/ftp-stuff/mural.pdf