HOL Developed and HOL Used: Interconnected Stories of Real-World Applications

Michael Norrish

July 2018
Cambridge Context in 1994—People

Recently finished/departed PhDs:

- Richard Boulton (efficient theorem-proving)
- Victor Carreno (real-time systems)
- Jim Grundy (refinement, window inference)
- Monica Nesi (process calculi)
- John Van Tassel (VHDL)
- John Harrison (real numbers, analysis)
... dreaming spires
My Cohort

Fellow PhD students:

- Mark Staples (refinement calculus in Isabelle/ZF)
- Don Syme (theorem-proving for operational semantics)
Starting a cl.cam.ac.uk PhD in 1994

Very flexible (more so than modern PhDs?)

- Don Syme changed topic completely after a year

Simultaneously gentle, and “sink-or-swim”:

- Mike suggested C as PhD topic as I got to grips with HOL
- I had a lot to learn
Cambridge Context in 1994—HOL

Powerful system moving beyond hardware verification applications

General purpose tooling:
- Inductive definition package
- Data type definition package
- Arithmetic decision procedures

Theorem-proving for operational semantics builds on all of these
My PhD

Almost entirely as a HOL user:

- mechanised an operational semantics for C (as per 1989 standard)
- proved some meta-theorems

Very much in vein of contemporary work applying HOL to operational semantics.

Examined by Tom Melham and Andy Gordon.
JRF and post-PhD Freedom

Won a Junior Research Fellowship at St. Catharine’s College
Could not muster much enthusiasm for C
HOL’s Continuing Development

Large ESPRIT project, “Prosper” (led by one Tom Melham) employs HOL’s then principal developer, Konrad Slind in Cambridge.

He and Ken Friis Larsen work on port from SML/NJ to Moscow ML

- Result is hol98; first release

  Athabasca-1

I attend various Prosper meetings and develop “opinions”.
Konrad’s openness to contributions lets me

- add a record type definition principle;
- completely rework HOL’s parsing and pretty-printing infrastructure;
- change the representation of numerals (from “unary” to binary scheme);
- name the relevant release series *Taupo*
Combining systems, continues to attack “hardware”-ish problems:

- With Ken Friis Larsen, integrates BDD package to allow CTL model checking (and other applications)
- Hardware description languages with Daryl Stewart
- First moves on ACL2 connections with Mark Staples
- (Later) Hardware synthesis with Juliano Iyoda
Mike and HOL: ARM

In 2000, Mike hired Anthony Fox on an ARM verification project

- joint work with Graham Birtwistle (Leeds), and support from ARM

This research project has been incredibly fruitful:

- Theorem-proving at scale ...
- ... leading to numerous real-world applications
Evaluation in the Logic

During visit from France, Coq developer Bruno Barras implements work-horse CBV_CONV (later just “EVAL”).

Critical tool for in-logic validation/execution of models

- Given time and expertise, custom tools could do sophisticated things
- Being able to type EVAL “f arg” to explore behaviours is an immense productivity boost
More Operational Semantics

HOL’s definitional tools scaled (scale) beautifully.

From tutorial examples (combinatory logic):

```
val (redn_rules, _, _) = Hol_reln `
  (!x y f. x --> y ==> f # x --> f # y) \/
  (!f g x. f --> g ==> f # x --> g # x) \/
  (!x y. K # x # y --> x) \/
  (!f g x. S # f # g # x --> (f # x) # (g # x))`;
```
More Operational Semantics

HOL’s definitional tools scaled (scale) beautifully.

To my C semantics
(one of many rules about assignment):

```plaintext
{hypotheses = [],
    side_conditions = [
        ('convert_val (strmap s) (v0,t0) (v,lhs_t) /
          (ok_refs = \x. x IN (se_affects (a, v)) => mb x \| 0) /
          (se’ = ref_map_fupd (\rm. BAG_DIFF rm ok_refs) se0) /
          (se = add_se (a, v) se’) /
          (resv = ECompVal v lhs_t)
          \/
          (!v. ~convert_val (strmap s) (v0, t0) (v, lhs_t)) /
          (resv = UndefinedExpr) /
          (se = se0)"
    ],
    (* --------------------------------------------------------------------- *)
    conclusion = "\^mng (mExpr (Assign CAssign (LVal a lhs_t)
      (ECompVal v0 t0)
      mb)
      se0) s (s, ^ev resv se)"
},
```
More Operational Semantics

HOL’s definitional tools scaled (scale) beautifully.

To ARM:

```haskell
val EXEC_INST_def = Define`
  EXEC_INST (ARM_EX (ARM reg psr) ireg exc)
    (dabort_t:num option) data cp_interrupt =
    if ~(exc = software) then
      EXCEPTION (ARM reg psr) exc
    else
      let ic = DECODE_INST ireg
      and (nzcv,i,f,m) = DECODE_PSR (CPSR_READ psr)
      in
      if ~CONDITION_PASSED nzcv ireg then
        ARM (INC_PC reg) psr
      else let mode = DECODE_MODE m in
      if (ic = data_proc) \ (ic = reg_shift) then
        DATA_PROCESSING (ARM reg psr) (CARRY nzcv) mode ireg
      else if ic = mla_mul then
        MLA_MUL (ARM reg psr) (CARRY nzcv) mode ireg
      else if ic = br then
        BRANCH (ARM reg psr) mode ireg
      else if (ic = ldr) \ (ic = str) then
        (LDR_STR (ARM reg psr) (CARRY nzcv) mode
         (IS_SOME dabort_t) (HD data) ireg).state
        (IC_SOME_dabort_t) (ALL data) ireg).
```
More Operational Semantics

HOL’s definitional tools scaled (scale) beautifully.

To TCP(?!):
Network Semantics

With **Peter Sewell** and Keith Wansbrough:

- Showed that HOL could handle large detailed semantics
  - first UDP and then TCP
  - both definitions, and generation of theorems in a novel style

- Developed custom tooling (the real HOL strength) to validate semantics against sniffed traces
TCP Work Driving HOL Development

Large terms, large theorems, large simplification sets...

Leading to:

- Another kernel implementation (more efficient with large numbers of bound variables)
  - suitably opaque & well-designed term API
- Dictionaries / trees in place of lists in various places

+ efficient evaluation...
Portability + Scalability = Better Tools

While a Cambridge post-doc, Scott Owens ports HOL to Poly/ML

- working with Sewell on hardware memory models
- fantastic speed-boost
- forces cleaner code
- allows powerful tools
Extending the HOL Diaspora

In 2003, I moved to Canberra.

HOL contributions came from

- Cambridge (Mike, students, postdocs)
- Oxford (Joe Hurd, Ashish Darbari)
- Australia (me and some students)
- USA (Konrad Slind, Peter Homeier, Joe Hurd)
- ...

A small, effective and harmonious developer community
Other Subsequent Work

Indirectly using C expertise:

▶ wrote “parser” tool to load seL4 C source code into Isabelle for verification project at NICTA (now Data61)

▶ HOL + ARM model allows for post hoc validation of this down to binary level

With Aditi Barthwal:

▶ formalisation of theory of context-free languages and parsing
  ▶ later useful in CakeML
Still to Come

Yet more operational semantics:

▷ $\mu$VM project with Blackburn, Hosking and Moss

More HOL development:

▷ broader visibility (github) 99% a good thing
▷ responsiveness to demands of major applications (i.e., mostly CakeML)
▷ learning lessons from Isabelle’s more extensive engineering
Mike

- Had a massive influence on my research career
- An energising emphasis on combining rigour with real-world applications
- Built a system; more importantly built community around it