Making HOL lots better

• The HOL4 proof assistant is an internationally used theorem-proving system (user pop'n: ~100?)

• It's used to ...

• ... prove mathematical theorems

• ... verify hardware & software
But...

HOL is the last in a line of programs based on an ancestor written in the 1970s

It shows...
Interaction window
And it's slower than it should be

- HOL is written in SML (not up for negotiation)
- Uses “inefficient” Moscow ML compiler
- Much faster (i.e., as good as C) compiler MLton available
In Four Words

- **Port**: move HOL kernel code to MLton compiler
- **Embed**: provide top-level loop by embedding customised Hamlet interpreter
- **Redesign**: theory mechanism (HOL's “separate compilation”), needs complete reworking
- **Plug-in**: connect redesigned HOL to Eclipse (building on existing Proof General code)
What lies ahead?
How much work is this anyway?

• Need to be happy with hacking small amounts of SML code

• Will get a chance to design a work-process (edit-compile cycle etc)

• Won't need to write very much new code, design complicated algorithms etc.

• For final link-up to Eclipse, will need Java skills