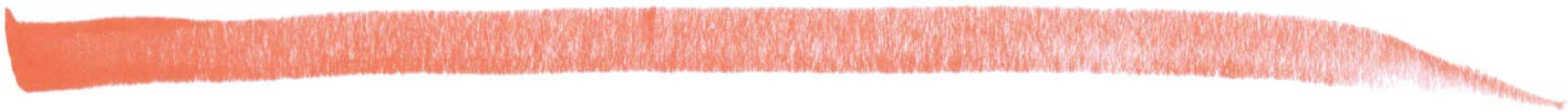


Brokers and Web-Services for Automatic Deduction: a Case Study

Claudio Sacerdoti Coen <sacerdot@cs.unibo.it>
Department of Computer Science
University of Bologna (Italy)

Stefano Zacchiroli <zack@di.ens.fr>
Department of Computer Science
École Normale Supérieure de Paris (France)

Outline

- Motivations
- HELM's Proof Assistant
- H-Bugs
  - architecture
  - a brief demo
  - implementation highlights (maybe)
  - future work

Motivations

Web-Service approach at software development . . .

- helps in getting rid of a wide range of software incompatibilities
- is (hopefully) granted to have longevity

. . . thus the WWW is moving . . .

from a disorganized repository of human-understandable HTML documents

to a disorganized repository of application exchanging XML documents

Motivations

The open challenge:

- Provide a set of stable and reliable services over this disorganization

What about solutions? . . . none :-(

- Just an useful architecture / widespread idea:
  add an additional intermediate level of stable services and call them brokers

HELM's Proof Assistant

Pros

- document centric
- exploits the HELM distributed library (≈ 40000 theorems/defs)
- modular, based on the HELM Web-Services (via XML + OMDoc)
- XML based GUI (XSLT stylesheets, MathML)
- a wonderful name: TODO (gTopLevel? CICIIDE?)

HELM's Proof Assistant

Cons

- Beginners' Cons
  - get lost in the available tactics
  - focus shouldn't be on interface issues, but rather on formal proof development
- Experts' Cons
  - computational expensive / resource consuming tactics
  - too low exploitation of the HELM library

An improvement

A suggestion engine that . . .

- . . . works in background
- . . . notify the user out-of-band
- . . . is as much re-usable as possible

Ωmega-Ants? H-Bugs!

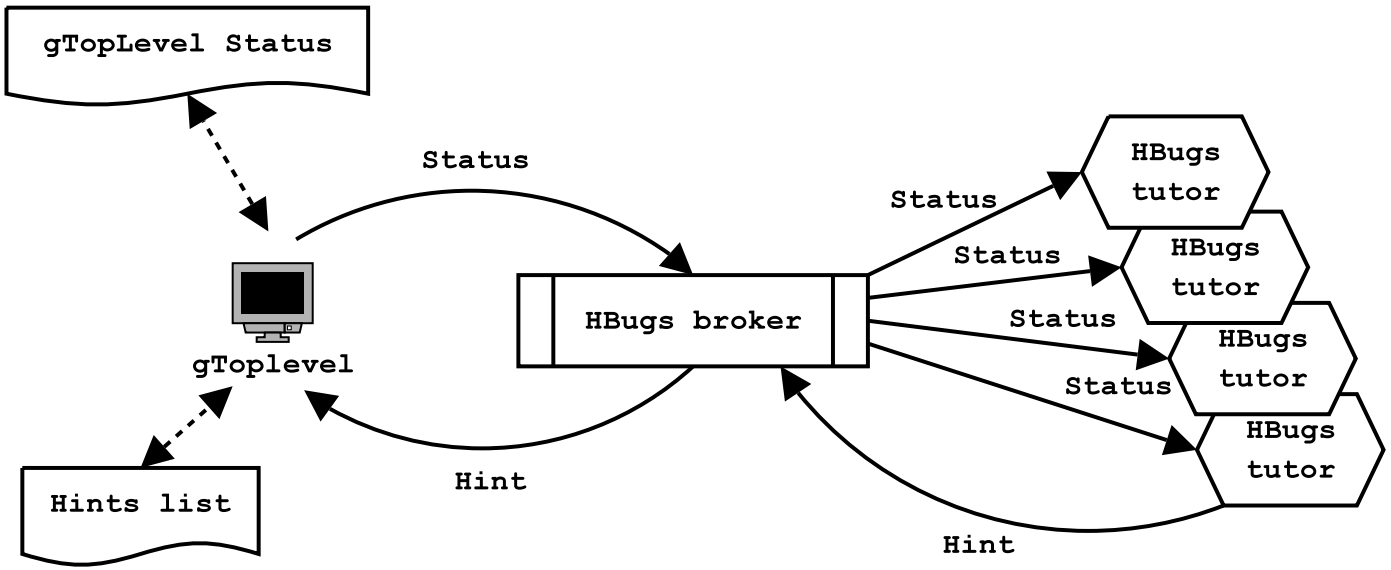
H-Bugs Actors

clients Software components able to produce proof status consuming hints

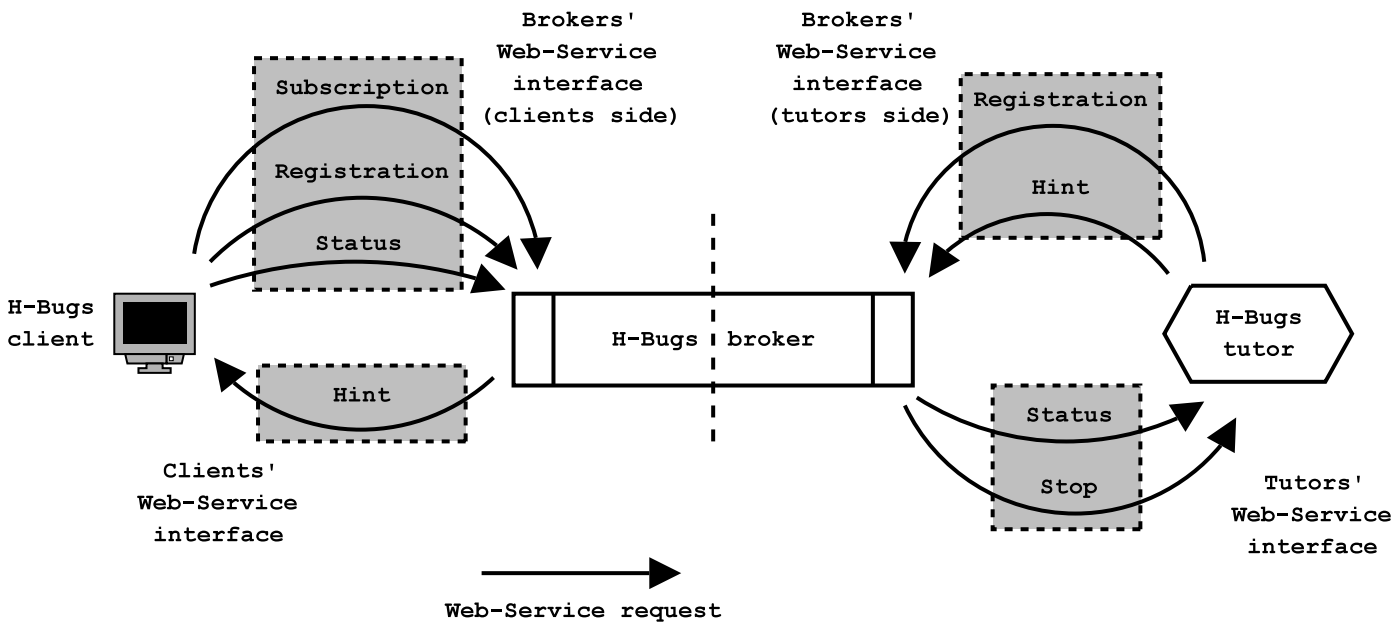
tutors Software components able to consume proof status producing hints

brokers Software components which act as intermediaries decoupling clients and tutors

H-Bugs Architecture



H-Bugs Web-Services



H-Bugs Messages

messages.dia

H-Bugs Tutors

I. Tutors for beginners

1. Assumption thesis equivalent to one of the hypotheses
2. Contradiction reductio ad absurdum
3. Simmetry commutative property over equalities
4. Left/Right/Exists/Split/Reflexivity/Constructor inductive type constructors application

H-Bugs Tutors

II. Tutors for computationally expensive tactics

1. Ring associative and commutative rewritings over ring structures

2. Fourier solves linear inequation systems (over R)

III. Intelligent tutors

1. Search pattern Apply search for applicable theorems from the HELM library (search + unification "attempts")

H-Bugs Demo

Exercise 1.

Let x be a generic real number. Using the HELM proof-engine, prove that:

$$x = \frac{(x+1)^2 - 1 - x^2}{2}$$

H-Bugs Tutors' Implementation

All tutors . . . :

- . . . are multi-threaded
- . . . present a Web-Service interface
- . . . perform a lot of similar operations (decoding proof status, encoding hints, parse and submit HTTP requests, . . .)

is not that difficult, but . . .
requires a lot of duplicated code/work :-(

H-Bugs Tutors' Implementation

To avoid the repetitive task of implementing a tutor each time we develop a new tactic:

I. we have written a generic tutor implementation

II. we instantiate it with each available tactic starting from an XML signature of each tutor

```
<tutor source="symmetry_tutor.ml">
  <addr>127.0.0.1</addr>
  <port>50004</port>
  <tactic>EqualityTactics.symmetry_tac</tactic>
  <hint>Hbugs_types.Use_symmetry_Luke</hint>
  <hint_type>Use Symmetry Luke</hint_type>
  <description>Symmetry tutor</description>
  <environment_file>symmetry.environment</environment_file>
</tutor>
```

H-Bugs Future Work

A lot!

- implement more (Intelligent) tutors
- interfacing of CASs (Field tactic?)
- non-hint (warnings?)
- hints rating
- MONET integration ontologies, reuse
- work on the GUI!

References

- A. Asperti, F. Guidi, L. Padovani, C. Sacerdoti Coen, I. Schena. Mathematical Knowledge Management in HELM. In Annals of Mathematics and Artificial Intelligence, 38(1): 27–46, May 2003.
- C. Benzmüller, M. Jamnik, M. Kerber, V. Sorge. Agent-based Mathematical Reasoning. In A. Armando and T. Jebelean (eds.), Electronic Notes in Theoretical Computer Science, (1999) 23(3), Elsevier.
- The MONET Consortium, MONET Architecture Overview, Public Deliverable D04 of the MONET Project.
  `http://monet.nag.co.uk/cocoon/monet/publicsdocs/monet-overview.pdf`
- S. Zacchiroli. *Web services per il supporto alla dimostrazione interattiva*, Master Thesis, University of Bologna, 2002.

That's all Folks!