Ideas and dreams for Isabelle

Makarius Wenzel
TU München

August 2009
Adequate interaction models

- Overcoming the TTY model (read-eval-print loop)
- Overcoming two-buffer (or three-buffer) model (Proof General and clones)
- Overcoming Emacs — use jEdit (or Netbeans, or ???)
- Towards parallel and asynchronous interactive proof checking
Proof document structure

theory C imports A B begin

inductive path for rel :: α ⇒ α ⇒ bool where
  base: path rel x x
| step: rel x y ⟹ path rel y z ⟹ path rel x z

theorem example: fixes x z :: α assumes path rel x z shows P x z
using assms
proof induct
  fix x show P x x ⟨proof⟩
next
  fix x y z assume rel x y and path rel y z
  moreover assume P y z
  ultimately show P x z ⟨proof⟩
qed
end
theory C imports A B begin

inductive path for rel :: α ⇒ α ⇒ bool where
  base: path rel x x
| step: rel x y ⇒ path rel y z ⇒ path rel x z ⟨internal proof⟩

theorem example: fixes x z :: α assumes path rel x z shows P x z
  using assms
  proof induct
    fix x show P x x ⟨proof⟩
  next
    fix x y z assume rel x y and path rel y z
    moreover assume P y z
    ultimately show P x z ⟨proof⟩
  qed

end
Theory libraries — formal repositories

• Overcoming proof “scripts” — proper documents
• Significantly improved presentation and browsing (XHTML/CSS, PDF)
• Reasonable queries, based on formal and informal content
• Integrating theory loader with distributed version control (Isabelle/Mercurial)
  – continuous proof checking, based on unique changeset ids
  – multi-author editing
  – formalized Wiki
**Scala/JVM system integration**

**Conceptual view:**
- Editor: JVM
- Document model
- Isabelle: SML

- bridge SML — Scala/JVM
- support GUls, IDEs, application servers etc.
- advanced document model: parallel checking, asynchronous interaction

**Implementation view:**
- Editor: JVM
- Scala
- Isabelle: SML

- public API, private protocol
- integral part of future Isabelle distributions