User Interfaces for Theorem Provers
August 21th 2006
Seattle, Washington, USA

PLATΩ: A Mediator between Text-Editors and Proof Assistance Systems
Motivation

• Evolution of computer-supported mathematics
• Low user-friendliness of actual proof assistants
• Interfacing scientific text-editors instead of yet another proof assistant GUI
Mediation Problem

- session management
- syntax analysis
- global transformation
- propagation of changes
- service interaction

Proof Language
- flexible semantic annotation

Service Menus
- context-sensitive interaction

- maintenance of consistent versions

Develop. Graph Representation
- structured math. theories

ΩMEGA
- logical structure
- sequents + focus
- proof planning

Task Layer Representation

Core
Proof Language

- Semantic annotation of natural language
  - macros in the text-editor
  - individual layout style
- Designed to support
  - textual structure of proofs
  - flexible and multiple positioning
  - underspecification
  - alternative proof attempts
- Parameterized over
  - languages for formulas, definitions and references
• Global Transformation
  - normalizing the flexible semantic annotation
  - separating proofs from theory knowledge
  - transforming linear proofs into the treelike datastructure

Proof Language (PL)
- theorems (flexible)
- proofs (flexible, linear)
- natural language text

Intermediate Language (IL)
- proofs (rigid, linear)

Development Graph Language (DL)
- theorems (rigid)

Task Language (TL)
- proofs (rigid, tree)
Maintenance of Consistency

- Propagation of Changes
  - semantic-based difference computation
  - efficient transformation of differences
  - preserving partial verifications in the proof datastructure

Proof Language (PL)
- theorems (flexible)
- proofs (flexible, linear)
- natural language text

Intermediate Language (IL)
- proofs (rigid, linear)

Development Graph Language (DL)
- theorems (rigid)

Task Language (TL)
- proofs (rigid, tree)

Propagation of Changes

TEXT EDITOR

Δ PL

Δ IL

Δ DL

Δ TL

PROOF ASSISTANT
Menu Request

- user selects object in the text-editor
- look up all related objects in the maptable of the transformation
- request menu from the proof assistance system for these objects
- display the menu in the text-editor
• Action Execution
  – user selects an action and its arguments in the menu
  – evaluate this action in the proof assistance system
  – evaluation result:
    • menu differences (nested evaluation)
    • document differences (top level evaluation)
  – propagation of changes
PLATΩ : System Demo

- Initializing a session
- Uploading a document
- Patching a document
- Requesting a menu
- Executing a menu action
- Closing a session

- XML-RPC server as interface for the text-editor
- Connection to the proof assistant in Lisp
- Operating as online webservice or local plugin
This theory defines the basic concepts and properties of the Theory of Simple Sets.

**Definition [Type of Individuals]**
First of all we define the type \( i \) of individuals.

**Definition [Type of Sets]**
Then we define the type set of sets of individuals.

**Definition [Function in]**
The function \( (i \times set) \rightarrow o \) takes an individual and a set and tells whether that individual belongs to this set.

**Definition [Function subset]**
The function subset \( (set \times set) \rightarrow o \) takes two sets and tells whether the first set is a subset of the second set.

**Axiom [Definition of subset]**
The function subset is defined by \( \forall U, V. \ (U \subseteq V) \iff \forall x. \ x \in U \ implies \ x \in V \).

**Definition [Function seteq]**
The function seteq \( = \) takes two sets and tells whether both sets are equal.

**Axiom [Definition of seteq]**
The function seteq is defined by \( \forall U, V. \ U = V \iff (U \subseteq V) \land (V \subseteq U) \).

**Definition [Function union]**
The function union \( (set \times set) \rightarrow set \) takes two sets and returns the union of both sets.

**Axiom [Definition of union]**
The function union is defined by \( \forall U, V, x. \ x \in (U \cup V) \iff x \in U \lor x \in V \).

**Definition [Function intersection]**
The function intersection \( (set \times set) \rightarrow set \) takes two sets and returns the intersection of both sets.

**Axiom [Definition of intersection]**
The function intersection is defined by \( \forall U, V, x. \ x \in (U \cap V) \iff x \in U \land x \in V \).
Theory [Distributivity in Simple Sets]

Context:
We refer to the definitions and axioms of the theory $\rightarrow$ [Simple Sets].

**Theorem [Distributivity of intersection]:**
It holds that $\forall A, B, C. \ (A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))$.

**Proof:**
We prove the equality of these sets by proving the following two subset relations:

1. $(A \cap (B \cup C)) \subseteq ((A \cap B) \cup (A \cap C))$
2. $((A \cap B) \cup (A \cap C)) \subseteq (A \cap (B \cup C))$
PLATΩ : System Demo

We prove the equality of these sets by proving the following two subset relations:

1. (pl-subgoal|SUBGOAL1|pl-formula|FORM6|Fsubset|Fintersection|V|A)|(Funion|V|B)|V|C) (Funion|Fintersection|V|A)|V|B)|(Fintersection|V|A)|V|C))

2. (pl-subgoal|SUBGOAL2|pl-formula|FORM7|Fsubset|Funion|Fintersection|V|A)|V|B)|V|C) (Fintersection|V|A)|V|B)|V|C))
PLATΩ : System Demo
PLATΩ : System Demo
Theory [Distributivity in Simple Sets]

Context:
We refer to the definitions and axioms of the theory \( \rightarrow \) [Simple Sets].

Theorem [Distributivity of intersection]:
It holds that \( \forall A, B, C. \ (A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C)) \).

Proof:
We prove the equality of these sets by proving the following two subset relations:
(1) \( ((A \cap (B \cup C)) \subseteq ((A \cap B) \cup (A \cap C)) \)
(2) \( ((A \cap B) \cup (A \cap C)) \subseteq (A \cap (B \cup C)) \)

We start by proving the first subgoal.
By the \( \rightarrow \) [Definition of subset] we need to prove
\( \forall x. \ x \in (A \cap (B \cup C)) \rightarrow x \in ((A \cap B) \cup (A \cap C)) \).
Assume \( x \in (A \cap (B \cup C)) \).
We start by proving the first subgoal.

\[
\langle \text{pl-proof} \rangle (\text{tuple} \_\text{key} \ | \ \text{PROOF2} \_\text{for} \ | \ \text{SUBGOAL1})
\]

\[
\langle \text{pl-fact} \rangle \text{FACT1}
\]

\[
\langle \text{pl-by} \rangle \text{BY1} \ | \ \text{By the} \ (\text{pl-reference} \ | \ \text{REF2} \ | \ \langle \text{L} \ | \ \text{Definition of subset} \rangle) \text{ we need to prove}
\]

\[
\langle \text{pl-obtain} \rangle \text{OBTAIN1} \ | \ (\text{pl-formula} \ | \ \text{FORM8} \ | \ \langle \text{forall} \ \langle \text{B} \ | \ \langle \text{V} \ | \ \text{x} \rangle \rangle \ | \ \langle \text{Fimpl} \rangle
\]

\[
\langle \text{Fin} \ | \ \langle \text{Fintersection} \ | \ \langle \text{V} \ | \ \text{A} \rangle \rangle \ | \ (\text{Finunion} \ | \ \langle \text{V} \ | \ \text{B} \rangle \ | \ \langle \text{V} \ | \ \text{C} \rangle) \rangle)
\]

\[
\langle \text{Fin} \ | \ \langle \text{V} \ | \ \text{x} \rangle \ | \ (\text{Finunion} \ | \ \langle \text{Fintersection} \ | \ \langle \text{V} \ | \ \text{A} \rangle \ | \ \langle \text{V} \ | \ \text{B} \rangle) \rangle \ | \ (\text{Finunion} \ | \ \langle \text{Fintersection} \ | \ \langle \text{V} \ | \ \text{A} \rangle \ | \ \langle \text{V} \ | \ \text{C} \rangle) \rangle) \rangle)
\]

\[
\langle \text{pl-assumption} \rangle \text{ASSUMPTION1}
\]

\[
\text{Assume} \ (\langle \text{pl-assume} \rangle \ | \ \text{ASSUME1} \ | \ (\text{pl-formula} \ | \ \text{FORM9} \ | \ (\text{Fin} \ | \ \langle \text{V} \ | \ \text{x} \rangle \ | \ (\text{Fintersection} \ | \ \langle \text{V} \ | \ \text{A} \rangle \ | \ (\text{Finunion} \ | \ \langle \text{V} \ | \ \text{B} \rangle \ | \ \langle \text{V} \ | \ \text{C} \rangle) \rangle) \rangle)
\]

\[
(\langle \text{pl-done} \rangle \langle \text{DONE1} \rangle)
\]
Theory [Distributivity in Simple Sets]

Context:
We refer to the definitions and axioms of the theory ↦ [Simple Sets].

Theorem [Distributivity of intersection]:
It holds that ∀A, B, C. \( (A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C)) \).

Proof:
We prove the equality of these sets by proving the following two subset relations:
(1) \((A \cap (B \cup C)) \subseteq ((A \cap B) \cup (A \cap C))\)
(2) \(((A \cap B) \cup (A \cap C)) \subseteq (A \cap (B \cup C))\)

We start by proving the first subgoal.
By the → [Definition of subset] we need to prove
∀x. \( x \in (A \cap (B \cup C)) \Rightarrow x \in ((A \cap B) \cup (A \cap C)) \).

Assume:
- Apply Definition of subset
- Apply Definition of set
- Apply Definition of union -
- Apply Definition of union +
- Apply Definition of intersection -
- Apply Definition of intersection +
Theory [Distributivity in Simple Sets]

Context:
We refer to the definitions and axioms of the theory ← [Simple Sets].

Theorem [Distributivity of intersection]:
It holds that ∀A, B, C. \((A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))\).

Proof:
We prove the equality of these sets by proving the following two subset relations:
(1) \((A \cap (B \cup C)) \subseteq ((A \cap B) \cup (A \cap C))\)
(2) \(((A \cap B) \cup (A \cap C)) \subseteq (A \cap (B \cup C))\)

We start by proving the first subgoal.
By the → [Definition of subset] we need to prove
∀x. \(x \in (A \cap (B \cup C)) \rightarrow x \in ((A \cap B) \cup (A \cap C))\).

Assume
Apply Definition of subset
Apply Definition of set=
Apply Definition of union -
Apply Definition of union +
Apply Definition of intersection - : Arguments: Compute Results
Apply Definition of intersection +
**Theory [Distributivity in Simple Sets]**

**Context:**
We refer to the definitions and axioms of the theory \(\rightarrow\) [Simple Sets].

**Theorem [Distributivity of Intersection]:**
It holds that \(\forall A, B, C. (A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))\).

**Proof:**
We prove the equality of these sets by proving the following two subset relations:
1. \(\{(A \cap (B \cup C)) \subseteq ((A \cap B) \cup (A \cap C))\}\)
2. \(\{((A \cap B) \cup (A \cap C)) \subseteq (A \cap (B \cup C))\}\)

We start by proving the first subgoal.

By the \(\rightarrow\) [Definition of subset] we need to prove:
\(\forall x. x \in (A \cap (B \cup C)) \rightarrow x \in ((A \cap B) \cup (A \cap C))\).

**Assume**

- Apply Definition of subset
- Apply Definition of set=
- Apply Definition of union -
- Apply Definition of union +
- Apply Definition of intersection - : Arguments: \(x \in A \land x \in (B \cup C)\)
- Apply Definition of intersection |
PLATΩ : System Demo
We start by proving the first subgoal.

\begin{verbatim}
(pl-proof)(tuple_key)PROOF2_for(SUBGOAL1)
  (pl-fact)FACT1
    (pl-by)BY1|By the (pl-reference)REF2 (L|Definition of subset)) we need
to prove
    (pl-obtain)OBTAIN1 (pl-formula)FORM8 (Fforall (B (V x))) (Fimpl
    (Fin (V x))(Fintersection (V A))(Funion (V B) (V C)))(Fin (V x)(Funion(Fin
    tersection (V A)(V B))(Fintersection (V A)(V C)))).
  (pl-assumption)ASSUMPTION1
    Assume (pl-assume)ASSUME1 (pl-formula)FORM9 (Fin (V x)(Fintersection
    (V A)(Funion (V B)(V C)))).
  (pl-fact)PL#3 It follows that (pl-derive)PL#1 (pl-formula)PL#2 (Fand (Fin
    (V x)(V A))(Fin (V x))(Funion (V B)(V C)))).
  (pl-done)DONE1)
\end{verbatim}
Theory [Distributivity in Simple Sets]

Context:
We refer to the definitions and axioms of the theory $\rightarrow$ [Simple Sets].

Theorem [Distributivity of intersection]:
It holds that $\forall A, B, C. (A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))$.

Proof:
We prove the equality of these sets by proving the following two subset relations:
(1) $((A \cap (B \cup C)) \subseteq ((A \cap B) \cup (A \cap C)))$
(2) $(((A \cap B) \cup (A \cap C)) \subseteq (A \cap (B \cup C)))$
We start by proving the first subgoal.
By the $\rightarrow$ [Definition of subset] we need to prove
$\forall x. x \in (A \cap (B \cup C)) \Rightarrow x \in ((A \cap B) \cup (A \cap C))$.
Assume $x \in (A \cap (B \cup C))$.
It follows that $x \in A \wedge x \in (B \cup C)$. 

article plato menus text roman 10 start
Related Work

• Automath, Mizar, Isar:
  – Balanced compromise between machine processable and human readable
  – Grammatical Framework: Framework to define grammar for an abstract and a concrete syntax

• PCOQ:
  – Schematic quasi-natural language output

• Nuprl, Clam, Omega/P.Rex:
  – Natural language processing technique to generate proof descriptions
Related Work (cont'd)

• Theorema:
  – Strictly separated formal and informal parts

• Mathlang:
  – Top-down from natural language
  – Use annotations for structure, no parser as well
  – Still even more far away from PAs

• ProofGeneral:
  – Top-down processing of documents
  – Documents are input format of PA rather than of some typesetting program.
Conclusion

- Interactive Mathematical Authoring with PLATΩ
  - Generic mediation between text-editors and PAs
  - Integrated support in text-editors as local plugin or online webservice
  - Parametric semantic annotation language
  - Automatic generation of the formal representation for the proof assistance system
  - Propagation of changes for efficiency and preservation of partial verifications
  - Context-sensitive service interaction
  - Maintenance of consistent formal and informal representations
Future Work

Text-Editors
- Emacs
- TeXmacs

Proof Assistants
- Omega
- Isabelle
- Coq
- Proof General

Natural Language
- MathLang
- Dialog