A New Framework for Reasoning Agents

(Towards Semantic MathWeb-SB)

Jürgen Zimmer

Talk by Christoph Benzmüller

Universität des Saarlandes, Germany.

This work is supported by the CALCULEMUS IHP Training Network HPRN-CT-2000-00102

Overview

- Limitations of the MathWeb Software Bus
- New Framework: Semantic MathWeb-SB
- Formal Descriptions of Reasoning Services
- First-order ATP Services
- Coordination of Reasoning Services

Motivation

Many specialized reasoning systems available:

- Deduction Systems:
  e.g., SPASS, Vampire, ..., SEM, Finder, ..., ΩMEGA, Isabelle
- Computer Algebra Systems (e.g., Maple, Mathematica, GAP)
- Mathematical Databases (e.g. MBase)
- Theory formation systems (HR)
- Further Tools: Proof Transformation, Proof Verbalization, . . .

Idea: Automated combination and coordination specialized systems

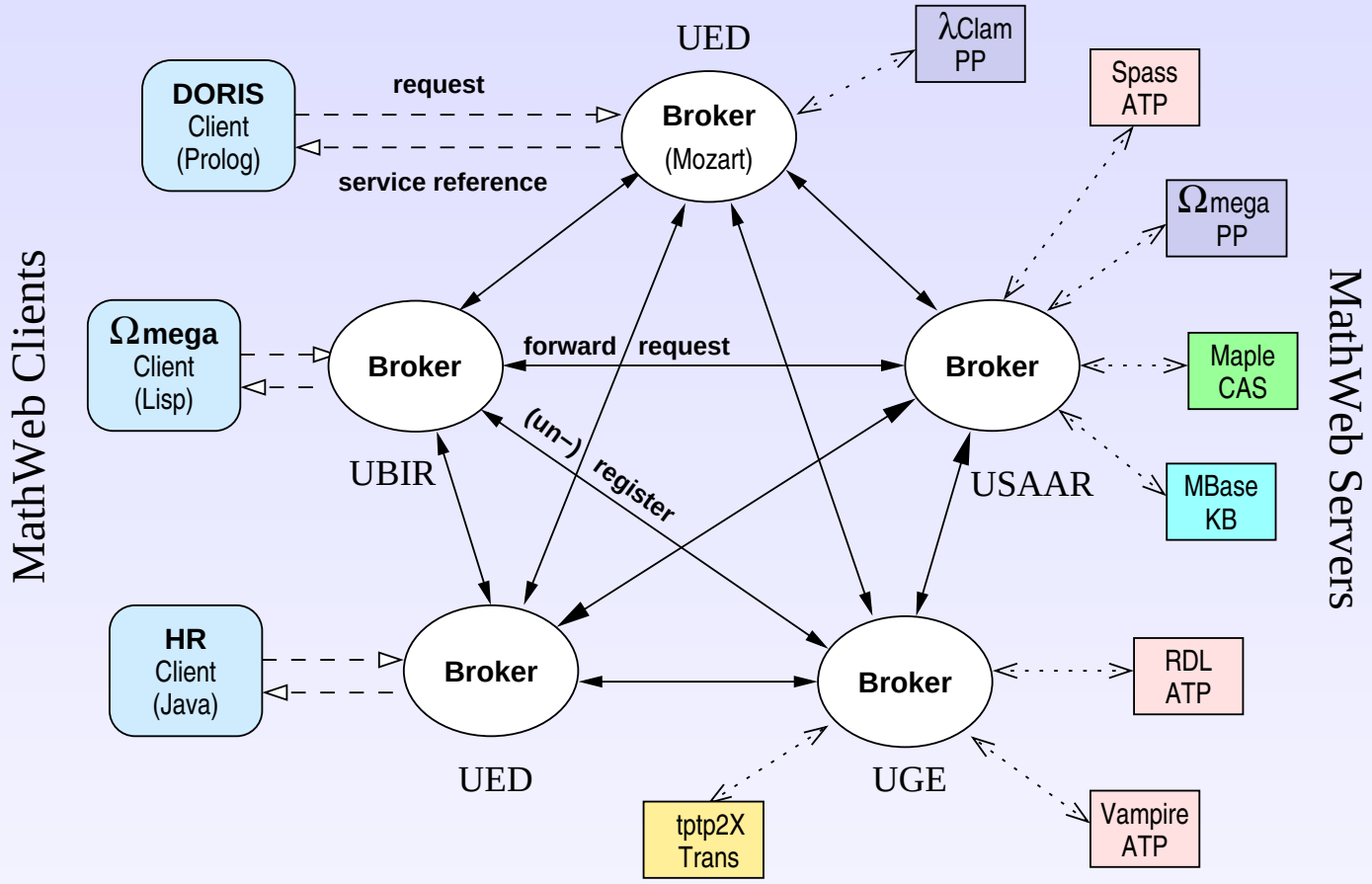
Why?: Reasoning systems are not open and only usable by experts

The MathWeb Software Bus

The MathWeb-SB combines reasoning systems on the system level:

- Connects $>$20 reasoning systems via common software bus.
- Based on standard languages for mathematical content (OPENMATH, OMDoc)
- Offers standard protocols (HTTP, XML-RPC)
- Binary distribution and sources available (GNU GPL)
- In CALCULEMUS RTN: MathWeb-SB supports CAS$\equiv$DS

The MathWeb Software Bus



Limitations of the MathWeb-SB

Despite its success, MathWeb-SB some limitations:

- Client applications still have to know
  - which reasoning system to use, and
  - how to access the system.
- User has to coordinate different reasoning systems to solve a problem.
- The MathWeb-SB is not designed for asynchronous communication.
- Technical Problems (OS, Firewalls, Proxies)

The New Framework

From MathWeb-SB to the Semantic MathWeb-SB:

- ... based on FIPA compliant agent platform
- ... agents offering reasoning services described in service description language

  $\Rightarrow$ looking at Semantic Web activity ($\rightarrow$ DAML-S)

  $\Rightarrow$ Cooperation with MONET and MathBroker project ($\rightarrow$ MSDL).
- ... a brokering mechanism for reasoning services.

First-order Theorem Proving Services

Every first-order proving service...

- ... accept standard problem formats TSTP and OMDoc
- ... is specialized on a particular domain
- ... should return a proof object in standard format

To build an ontology for first-order ATPs:

- Use work with Geoff Sutcliffe and Stephan Schulz on generic proof format and ATP states.

To obtain specialized services for brokering:

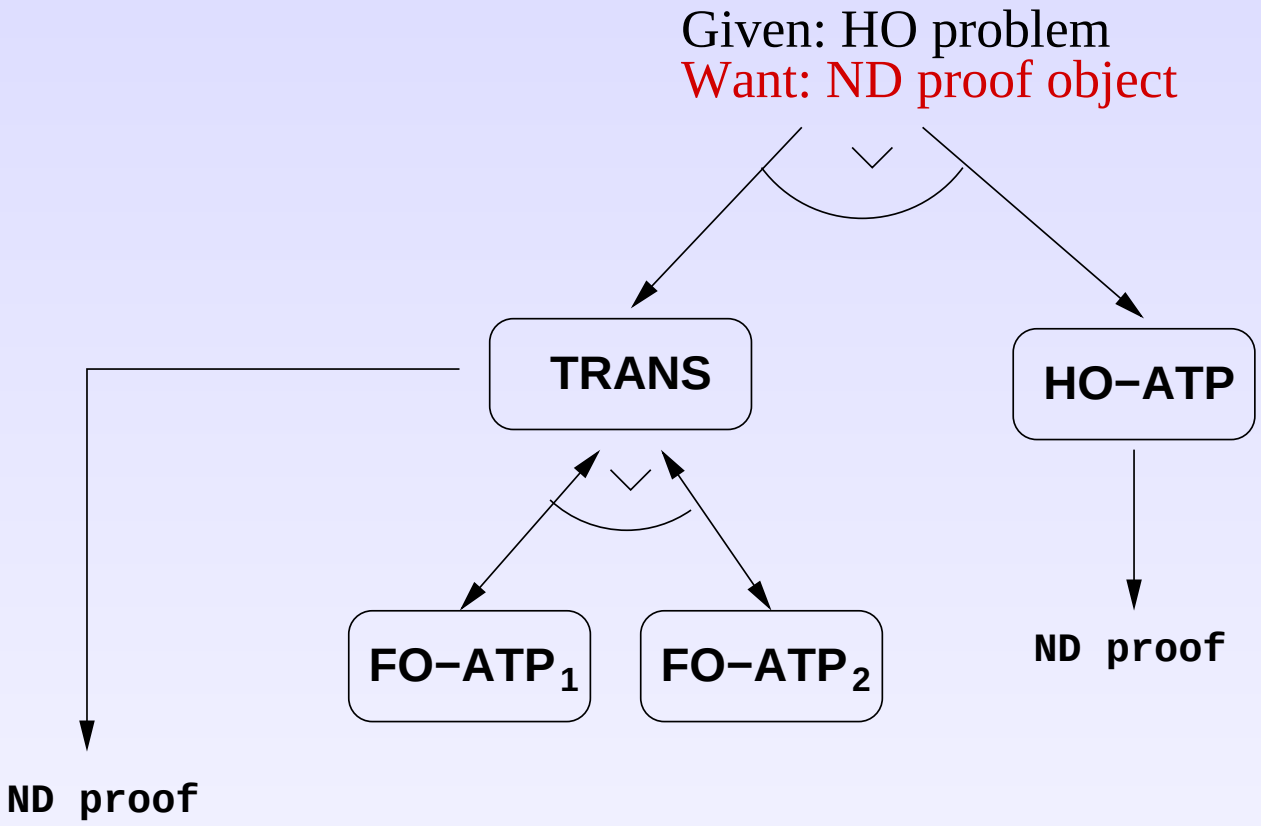
- Use work done on tuning ATPs towards problem domains in TPTP library.

An ATP Service in MSDL

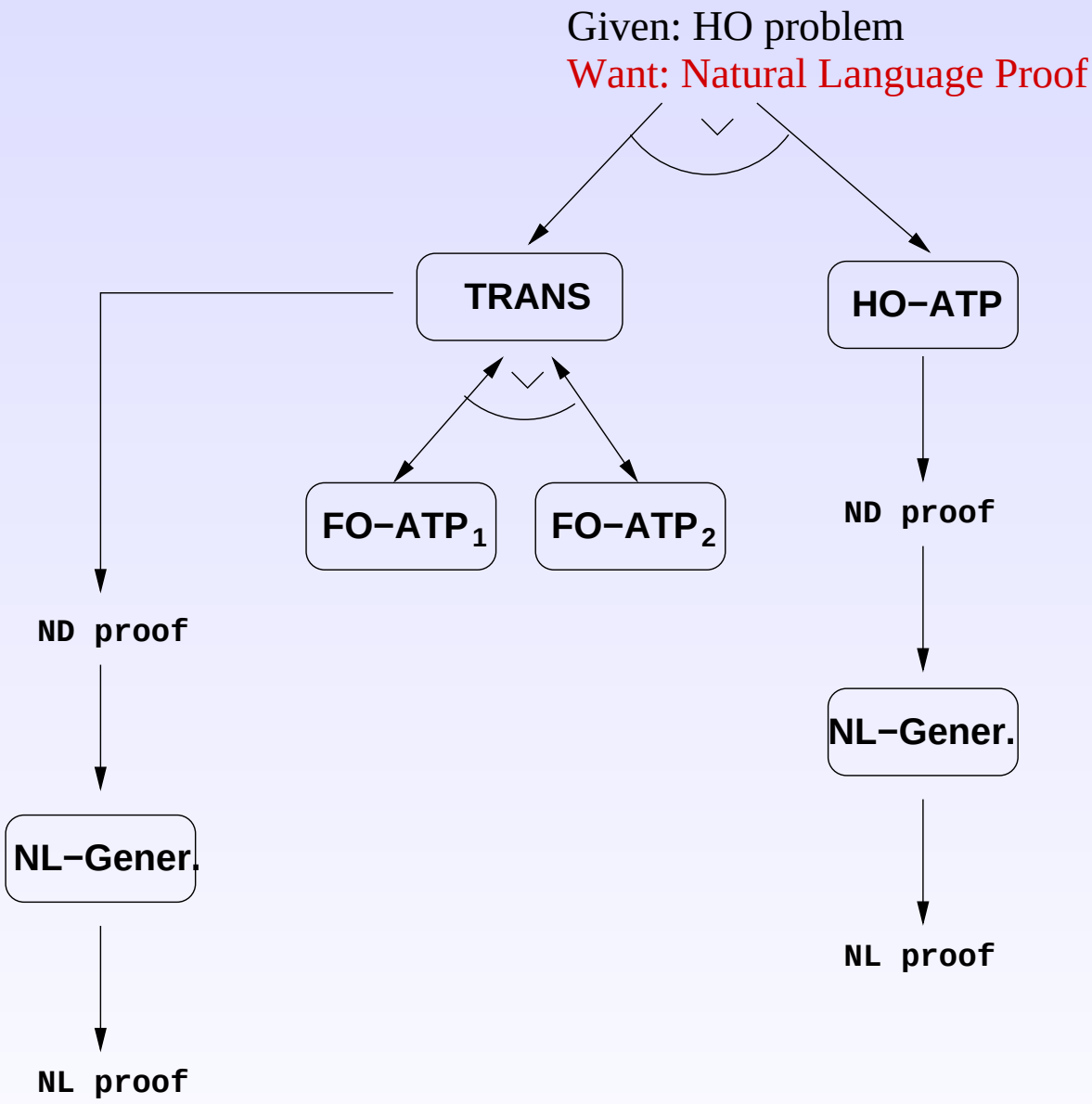
Service: SpassProver	
classification:	Classifi cation with Taxonomy of services or link to Ontology (→ QPQ) • → **first-order problem description**
service interface:	→ `fo-prover.wsdl`
implementation details:	Information about hardware, software (calculus, etc.)

first-order problem description	
input parameters:	name: $problem$, signature: ATP-Problem (DAML-S Class)
output parameters:	name: $result$, signature: ATP-Result (DAML-S Class)
pre-conditions:	$equational\_reasoning(problem)$ $\wedge\ Ax = axioms(problem)$ $\wedge\ C = conjecture(problem)$ $\wedge\ \forall a \in Ax.first\_order(a)$ $\wedge\ first\_order(C)$ $\ldots$
post-conditions:	$Ax \vdash_{FOL} C \ \Rightarrow has(result, proof\_object)$

Coordination of Mathematical Services



Coordination of Mathematical Services



Related Work

- MONET (Mathematics on the Semantic Web) and MathBroker Project: Brokering and coordination of CAS computations
- ETI (Electronic Tool Integration platform) Project: Brokering and coordination of verifi cation tools for real time systems and model checkers
- QSL Project: User assisted coordination of reasoning tools
- Grid and Semantic Grid: Sharing of computation resources
- QPQ (QED Pro Quo) Project: Repository of deductive software

Conclusion

To overcome limitations of the MathWeb-SB we propose

- ... a framework for reasoning agents offering (semantically described) services
- ... a semantic brokering and coordination mechanism

We started describing fi rst-order ATPs:

- using ongoing work on TSTP
- designing an ontology for proving services (incl. logics & calculi)

Future Work

To overcome problems with

- Knowledge retrieval from ATP users and developers.
- First descriptions of ATPs.
- Prototypical broker for specialized ATPs.
- Build up ontology for reasoning systems.
- Description of other reasoning systems (e.g., model generators).