Application for a DFG Research Grant

VERMATH

Distributed Mathematical Problem Solving

April 28, 2004

1 General Information (Allgemeine Angaben)

1.1 Applicants (Antragsteller)

This is a joint application by two partner universities: Saarland University (1 researcher) and International University Bremen (1 researcher). It is also based on substantial international co-operation with american and english universities (hence we use English for this document).

(A) Saarland University

1.1.1
- Serge R. Autexier, Dr.
- Senior researcher
- 2.10.1971, German and French
- German Research Centre for Artificial Intelligence (DFKI GmbH)
  Mailing address: Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
  Tel: +49 681 302-5073 Fax: +49 681 302-5076
  E-mail: `autexier@dfki.de`
- Private address: Großherzog-Friedrich Str. 73, 66111 Saarbrücken
  Tel.: +49 681 375620
- Curriculum vitae and list of publications (see appendix A)

1.1.2
- Christoph E. Benzmüller, Dr.
- Hochschuldozent (C2)
- 8.9.1968, German
- Universität des Saarlandes, Fachrichtung 6.2 Informatik
  Mailing address: P.O. Box 15 11 50, 66041 Saarbrücken, Germany
  Tel: +49 681 302-4574 Fax: +49 681 302-5076
  E-mail: `chris@ags.uni-sb.de`
- Private address: Am Landwehrplatz 6-7, 66111 Saarbrücken
  Tel.: +49 681 9659107
- Curriculum vitae and list of publications (see appendix B)

1.1.3
- Jörg H. Siekmann, Prof. Dr. (Ph.D.), grad. Ing.
- Professor
- 5.8.1941, German
- Universität des Saarlandes, Fachrichtung 6.2 Informatik
  Mailing address: P.O. Box 15 11 50, 66041 Saarbrücken, Germany
  Tel: +49 681 302-5276 Fax: +49 681 302-5076
  E-mail: `siekmann@ags.uni-sb.de`
- Private address: Robert-Koch Str. 6, 66119 Saarbrücken
  Tel.: +49 681 55447
- Curriculum vitae and list of publications (see appendix C)

(B) International University Bremen

1.1.4
- Michael Kohlhase, Prof. Dr.

- Professor for Computer Science
- 13.9.1964, German
- International University Bremen, School of Engineering and Science
  Mailing address: P.O. Box 750 561, 28725 Bremen, Germany
  Tel: +49 421 200-3140 Fax: +49 421 200-3103
  E-mail: `m.kohlhase@iu-bremen.de`

1. Private address: Mühlental 5, 28717 Bremen, Germany
   Tel: +49 421 6396849

2. Curriculum vitae and list of publications (see appendix D)

1.2 Topic (Thema) .... Distributed Mathematical Problem Solving

1.3 Code name (Kennwort) .... VERMATH

1.4 Scientific discipline (Fachgebiet und Arbeitsrichtung)

Artificial Intelligence, Deduction Systems, Distributed Systems

1.5 Scheduled duration (Voraussichtliche Gesamtdauer) ....4 years

1.6 Application period (Antragszeitraum) .... 2 years

1.7 Start of funding (Gewünschter Beginn der Förderung) ..... August 2004

1.8 Summary (Zusammenfassung)

The Internet provides a vast collection of data and computational resources. For example, a travel booking system combines different information sources, such as the search engines, price computation schemes, and the travel information in distributed very large databases, in order to answer complex booking requests. The access to such specialized travel information sources has to be planned, the obtained results combined and, in addition the consistency of time constraints has to be guaranteed.

We want to transfer and apply this methodology to mathematical problem solving and develop a system that plans the combination of several mathematical information sources (such as mathematical databases), computation engines (such as computer algebra systems), and reasoning processes (such as theorem provers or constraint solvers) to solve mathematical problems. The project will be based on the well-developed MathWeb-SB network of existing mathematical services, whose client-server architecture will be extended by advanced problem solving capabilities and semantic brokering of mathematical services. The current MathWeb-SB is already well established internationally with up to 500 theorems and small lemmata being submitted and processed per hour. These mathematical problems come from unification systems, semantic natural language processing as well as genuine mathematical theorems. We intend to describe

the mathematical services in a formal service description language. Reasoning over such service descriptions with for instance AI planning techniques, can then be used to find appropriate sequences of services to solve the particular mathematical problem at hand.

This is a joint project between the University of Saarbrücken (Prof. Siekmann) and the International University Bremen (Prof. Kohlhase).

2 State of the art, preliminary work (Stand der Forschung, eigene Vorarbeiten)

2.1 State of the art (Stand der Forschung)

Traditionally, mathematical problem solving systems for symbolic reasoning and symbolic computation, are stand-alone systems. Typical systems are the first-order automated theorem provers Otter [McC90], Spass [Wei97], or VAMPIRE [RV01] and computer algebra systems such as Mathematica [Wol99] or Maple [Red96]. A recent trend, however, is to gradually open the architectures of such systems. This is particularly the case for interactive theorem proving environments like NuPrl [ACE+00], HOL [GM93], , PVS [ORS92], Coq [Tea], and ΩMEGA [SBB+02]. Instead of a fixed and hardwired control, the idea is to out-source and exchange specialized mathematical services like decision procedures, simplification tools, mathematical databases, presentation tools, logic- and proof-transformation components, etc.

Several networks for such distributed mathematical service systems are currently under development. Prominent examples are our MathWeb-SB [ZK02] (which will be discussed in more detail in section 2.2), the Prosper Toolkit [DCN+00], or MetaPrl [HNC+03] (discussed in more detail in section 2.1.1). In these mathematical service networks client systems may employ service systems via the provided network infrastructure. Thereby communication is supported by respective communication protocols. A typical example is a theorem prover calling a computer algebra system to perform some computational task or a decision procedure, e.g. a SAT-solver, to decide the satisfiability of a sub-problem. Clearly, some mathematical tools in these scenarios generally qualify as client systems and as service systems simultaneously.

The brokering of mathematical service systems in such networks is still not well understood and several recent research projects try to address this problem. For example, the projects *Math-Broker* [SC01] and MONET [Con02] (see further details below) aim at the development of a service description language and a taxonomy of existing systems in order to improve the semantical brokering of computational services.

European research networks in which this research plays a major role are the CALCULEMUS Network[1] (which investigates the Integration of Computation and Deduction Systems) [Ben03], the MONET project[2], and the new MKM initiative[3] (for research in Mathematical Knowledge Management). Our group coordinates the CALCULEMUS Network and it is also part of the MKM

[1] http://www.eurice.de/calculemus
[2] http://monet.nag.co.uk/cocoon/monet/index.html
[3] http://monet.nag.co.uk/mkm/index.html

initiative as well as the MOWGLI consortium[4], which shall enable us to closely cooperate with these research initiatives.

2.1.1 Networks of mathematical service systems

The MathWeb Software Bus (MathWeb-SB) [ZK02] is a distributed system for the integration of *reasoning systems*, i.e., deduction systems and symbolic computation systems. It has been developed at Saarbrücken and Edinburgh and is described in more detail in section 2.2.

The PROSPER project [DCN+00] developed a technology to deliver formal specification and verification to system designers in industry. The central idea of the PROSPER project is that of a *proof engine* (a custom built verification engine) which can be accessed by applications via an application programming interface (API). This API supports the construction of CASE-tools[5] incorporating a user-friendly access to formal techniques. The API of PROSPER mainly allows the modularization of CASE tools and software systems and the inter-operation of these modules with a proof engine. However, the components of the PROSPER toolkit still use a proprietary protocol for communication.

MetaPRL [HNC+03] is a logical framework where multiple logics can be defined and related. In MetaPRL, mathematical knowledge can be exchanged between the different logic contexts. Furthermore, it is a system with support for interactive proving and automated reasoning. The development of the MetaPRL system is motivated by the enormous investment in systems like PVS, HOL, Coq, ELF, Nuprl, and others, which use different logics and different methodologies, but have a common goal and their results share fundamental mathematical underpinnings. MetaPRL tries to expose the logical foundations of these systems and to share the results between the systems. MetaPRLs incremental module construction allows logical re-use: logics can share a common core with properties that are inherited as the logics are extended.

2.1.2 Mathematical Services

The *MathBroker* project [SC01] develops a software framework for brokering symbolic computation services that are distributed among networked servers. The foundation of this framework is a language for describing the mathematical problems and a taxonomy of pre-defined mathematical problems. Servers register their problem solving capabilities with a "semantic broker" to which clients submit corresponding task descriptions. For a given task, the broker determines a service suitable to tackle that task and returns the description of that service to the client for invocation. Embedded into XML documents and interpreted by browser applets, service descriptions can act as interactive hyper-media interfaces for distributed mathematical applications.

One area of interest of the MONET project [Con02] are mathematical algorithms offered via web services that can be accessed from a wide variety of software packages. The challenge is to develop a framework in which such services can describe their capabilities in as much detail as is necessary to allow a sophisticated software agent to select a suitable service based on an analysis of the characteristics of a user's problem.

[4] `http://mowgli.cs.unibo.it`

[5] CASE: Computer Aided Software Engineering

Both the *MathBroker* and the MONET project have goals similar to the research proposed here. However, both projects deliver web services by existing standards and existing web technology, such as RDF [BG00] and WSDL [CCMW01]. Furthermore, both projects focus on computer algebra systems and numerical algorithms. In our research, we focus on the combination of heterogeneous reasoning systems and service tools, such as theorem provers, decision procedures, computer algebra systems, logic- and proof transformation services, and constraint solvers. In contrast to *MathBroker* and MONET we intend to build a framework on top of existing standards of multi-agent technology, such as the FIPA specification [Fip02]. We also want to replace the rather simple semantical brokering approach proposed in *MathBroker* by an advanced plan based semantical brokering mechanism: while service requests in *MathBroker* are directly linked to particular service systems whose semantical description matches the request, we propose *to plan* how available service systems can be suitably combined in order to cooperatively tackle a problem.

We shall build on our longterm experience in theorem proving and on the integration of heterogeneous reasoning systems in the MathWeb-SB. This is particularly important since the MathWeb-SB already provides the service systems that are required for the advanced theorem proving brokering we envisage. In order to ensure compatibility with *MathBroker* and the MONET project, however, we want to further improve our existing cooperation — in particular for the definition of a description language for mathematical services as proposed in the following section.

2.1.3 Capability Descriptions in Multi-Agent Systems

Distributed AI and multi-agent systems in particular use special description languages to describe the capabilities of intelligent agents or services provided by these agents. Two languages, the *capability description language* (CDL) [Wic99] developed at the University of Edinburgh and *The Language for Advertisement and Request for Knowledge Sharing* (LARKS) [SWKL01], seem of particular importance for our project. These languages combine a high expressivity with the possibility to reason on these capability descriptions, as the semantics of a capability is essentially described by Horn-clauses which allows the use of standard Prolog engines. LARKS descriptions can be based on heterogeneous ontologies formalise in the ontology language ITL [Gua91]

2.1.4 The Semantic Web Initiative

The Semantic Web Initiative [FHLW03] shares the vision of using common standards to annotate data on the web with meta-data such that it can be used and understood by machines. Several XML-based standards are emerging to express meta-data in the semantic web, such as the Resource Description Framework (RDF) [BG00], and to define ontologies and reason on them, such as the DAML+OIL [HvHPS01] and the Web Ontology Language (OWL) [SMVW02].

TRIPLE [SD02] is a query, inference, and transformation language for reasoning about RDF descriptions and provides access to external programs (like classifiers of description logics). Reasoning engines like TRIPLE will be interesting in our proposed research, because a broker for mathematical services will eventually need to reason about concepts and their relationships in a mathematical ontology (cf. section 3.1.1).

2.1.5 Content Languages for Mathematics

The OPENMATH language [CC98] provides a standard for representing mathematical objects and their semantics such that they can easily be exchanged between computer programs, stored in databases, or published in the worldwide web. While the original design was mainly for computer algebra systems, OPENMATH is now attracting interest from other areas of scientific computation as well as from many publishers of electronic documents with mathematical content. Our research group has many years of experience in the use of OPENMATH for the communication between deduction and computer algebra systems.

There is a strong relationship between OPENMATH and the *Content MathML* recommendation of the Worldwide Web Consortium [CIMP01] and the new versions of Content MathML and OPENMATH are equally expressive and formulae can be translated from one language into the other. Content MathML also allows semantic information encoded in OPENMATH to be embedded into a MathML structure. *Presentation MathML* deals principally with the presentation of mathematical objects, while OPENMATH is concerned with their semantic markup. Thus the two technologies may be seen as highly complementary.

The OMDoc language [Koh01] for advanced semantical markup of mathematical content has been developed by Michael Kohlhase. It is described in more detail in section 2.2.

2.2 Preliminary work (Eigene Vorarbeiten)

The MathWeb Software Bus. The MathWeb Software Bus (MathWeb-SB) [ZK02] is a distributed system for the integration of *reasoning systems*, i.e., deduction systems and symbolic computation systems. The main idea of MathWeb-SB is to extend existing reasoning systems, such as automated theorem provers (ATPs), computer algebra systems (CASs), model generators (MGs), and constraint solvers (CSs) with a generic interface that allows them to communicate over a common software bus. Figure 1 shows the current structure of the MathWeb-SB. Mathematical applications are integrated homogeneously as "servers" in a network whose central nodes are the so-called "MathWeb brokers". However, these brokers have currently very restricted capabilites, and they are essentially just yellow-page servers. When started, the servers register themselves to their local "broker". Brokers register and de-register dynamically to other brokers in the Internet or an Intra-net, which are called *remote brokers*. Client applications connect to a known broker and request access to a reasoning system. In case the requested system is available locally, i.e., it is offered by a local server, a new instance of the reasoning system is created and a reference to this instance is sent back to the client application. If a reasoning system is not offered by a local server, the broker forwards the request to all known remote brokers.

MathWeb-SB Client Applications. The MathWeb-SB was developed originally to integrate external reasoning systems into the ΩMEGA system [SBB+02] and to make these reasoning systems accessible for human users as well as for knowledge-based proof planning.

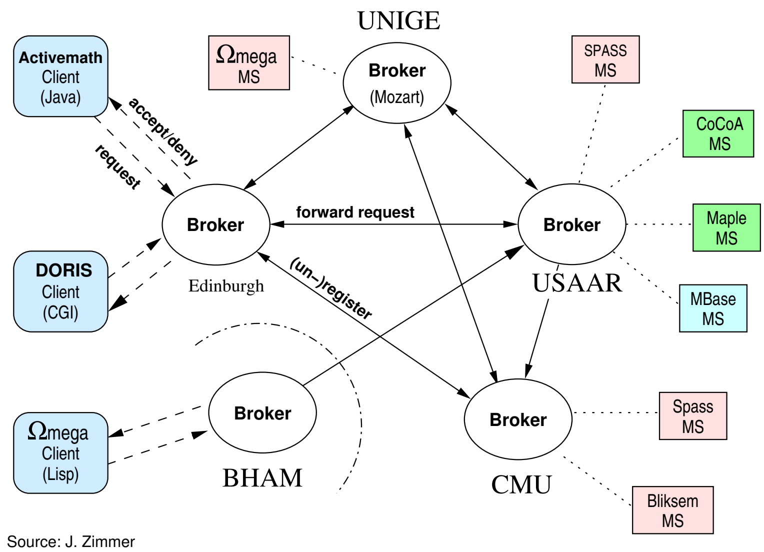

Figure 1: The MathWeb Software Bus

In the last years, we have further developed the MathWeb-SB into a stable and generic platform and many new reasoning systems are now integrated. There is also a growing number of client applications using the MathWeb-SB:

- The OMEGA and MBASE projects (Saarland University and International University Bremen, Germany)
- The MAYA project (Deutsches Forschungszentrum für Künstliche Intelligenz, Germany)
- The DREAM group of Prof. Alan Bundy (University of Edinburgh, Scotland)
- The research group of Dr. Manfred Kerber and Dr. Volker Sorge (University of Birmingham, England)
- Projects at Carnegie Mellon University, USA
- The DORIS system of Dr. Johan Bos (University of Edinburgh, Scotland)

The mathematical services available via MathWeb-SB are:

- Proof assistants: ΩMEGA, CORE, PVS.
- Mathematical database: MBASE.
- Theory formation system: HR.
- Constraint solver: $\mathcal{C}o\mathcal{SIE}$.
- Proof explanation system: *P. rex*; an external system developed by the ΩMEGAgroup [Fie01].
- Generalization module of LEARNΩMATIC.
- Proof planners: MULTI, λCLAM.
- Computer algebra systems: MAPLE, GAP, COCOA, MAGMA.
- Automated higher-order theorem provers: TPS, LEO.

- Automated first-order theorem provers: BLIKSEM, OTTER, PROTEIN, SPASS, VAMPIRE.
- Automated equational provers: E, EQP, WALDMEISTER.
- Model generators/checkers: SEM, FINDER, MACE, SATCHMO.
- Proof transformation system: TRAMP.
- Translators for OMDoc and TPTP.

Mathematical Services in the MathWeb-SB. The reasoning systems in the MathWeb-SB offer mathematical services, however, these services have not been formally specified. Client applications can only use a reasoning system of MathWeb-SB if they know the name of the system and if they have exact knowledge about the application programming interface (API) of the reasoning system. Sometimes, a client application has to know even the input syntax and other peculiarities of the underlying reasoning system (e.g, Maple syntax). This is too much of a burden in particular if the developer has no or only limited knowledge about the reasoning systems involved, as e.g. the DORIS system developed in the computational linguistics department of the University of Edinburgh. It is important to abstract from the implementational details of a reasoning system and to make it easier for client applications to use the services already available in the MathWeb-SB. One step towards this goal is the specification of (parts of) the functionality of reasoning systems as a mathematical service.

The OMDoc language. OMDoc [Koh01] is a markup format and data model for *O*pen *M*athematical *Doc*uments on the Web. OMDoc differs from other presentation-based approaches in that it concentrates on representing the *structure* of mathematical concepts and documents instead of their *presentation*. OMDoc is an extension of OPENMATH and content MATHML, by a markup for the document and theory level of mathematical documents. The document author specifies the semantics of a mathematical concepts with these mark-ups such that the consumer (an OMDoc reader or a mathematical software system) can use it. In general, the standard languages OPENMATH [CC98] and OMDoc [Koh01] are used as content languages for the client-server communication in the MathWeb-SB.

3 Goals and work schedule (Ziele und Arbeitsprogramm)

3.1 Goals (Ziele)

The reasoning systems currently integrated in MathWeb-SB have to be accessed directly via their API, thus the interface to MathWeb-SB is system-oriented. However, reasoning systems are used by applications that are not necessarily theorem provers, e.g. for the semantical analysis of natural language, small verification tasks, etc. The main goal of this project is twofold:

Problem-Oriented Interface: to develop a more abstract communication level for MathWeb-SB, such that general mathematical problem descriptions can be sent to the MathWeb-SB which in turn returns a solution to that problem. Essentially, this goal is to move from a *service* oriented interface to a *problem* oriented interface for the MathWeb-SB. This is a very old idea in the development of AI programming languages (early work included PLANNER and other languages driven by the problem matching of general descriptions). To this end the project is concerned with the development of an appropriate language and an ontology for the description of mathematical problems. The ontology will serve as the basis for the new interface of MathWeb-SB.

Advanced Problem Solving Capabilities: Second, the project aims at integrating even more advanced problem solving capabilities into MathWeb-SB. Typically, a given problem cannot be solved by a single service but only by a combination of several services. So far, the decomposition of a given mathematical problem into subproblems which are in the domain of the existing services is left entirely to the user of MathWeb-SB. Thus, the project is also concerned with developing enhanced solving capabilities that decompose the problem into subproblems such that their solution is within the range of a combination of existing services.

In order to support the automatic selection and combination of existing services, the key idea is as follows: the ontology developed in the first part of the project will be used for the qualitative description of MathWeb-SB services and *these descriptions will then be used as AI planning operators*, in analogy to todays proof-planning approach. We can then use AI planning techniques [CBE+92, EHN94] to automatically generate a plan that describes how existing services must be combined to solve a given mathematical problem.

In the current MathWeb-SB the notion of *service* denotes an existing reasoning system, like a theorem prover, a computer algebra system, a constraint solver, or others. We envision on the one hand to make the notion of a mathematical service more concrete by using fine-grained service descriptions. On the other hand, we also want to extend this notion by allowing a service to be also a reference to a publication that describes a mathematical technique without an available implementation. This allows us to further enhance the problem solving capabilities of the MathWeb-SB beyond problems that can only be solved by existing systems. The solution to these mathematical problems amounts to a description of how techniques known in the literature could be combined in order to solve the problem.

This provides also the basis for the use of MathWeb-SB as a tool for mathematicians and the theorem proving communities to query the literature of problem solving techniques, as soon as it is available in the upcoming semantic web for mathematics. The answer to such a query not only contains a list of publications, but also a description of how the described techniques can be combined.

3.1.1 Description Language for mathematical Tasks and Services

An XML-based Service Descriptions Language. A description of a mathematical service should not only describe the input and output parameters of a service but also, to a certain extent, the semantics of the service, i.e. the relationship between the input and the output. Similar to say a LARKS specification (cf. section 2.1.3), we intend to express the semantics of a service as constraints over the output and input parameters of a service using statements in a first-order logic fragment (for instance, horn-clauses).

Service descriptions should be readable and understandable by both humans and machines. The extensible markup language (XML) [Mar01] seems to offer currently the best compromise between human and machine-readability. XML is playing an increasingly important role in the exchange of a wide variety of data in the Internet and elsewhere. For this reason and to preserve compatibility with other projects like OPENMATH, OMDoc, MONET, and *MathBroker*, we intend to base our service description language fully on XML.

In Figure 2 we show how a simplified version of the specification of a first-order theorem proving service FO-ATP may look in an XML service description language.

```
<provingService name=" FO-ATP">
   <comment>
       Tries to prove the validity of a first-order
       theorem proving problem within the number of
       seconds given as a timeout.
   </comment>
   <input>
       <parameter name="problem" type=" ProvingProblem"/>
   </input>
   <output>
       <parameter name="result" type=" FO-ATP-Result"/>
   </output>

   <inConstraints>
       has_some_equality(problem), real_first_order(problem).
         .........
   </inConstraints>
</provingService>
```

Figure 2: Example XML specification of a first-order theorem proving service

The FO-ATP service expects a problem description as input. It always returns an object of type FO_ATP_Result (see below). The `inConstraints` of the service say that the service is most suitable if the problem has some equalities in it and also that it should really be a first-order problem (and not essentially propositional).

An Ontology for mathematical Service Descriptions. Figure 3 illustrates an ontology for first-order automated theorem provers, as it was used above. Such an ontology could, for instance, be defined in the ontology language OWL [SMVW02].

Our ontology consists of concepts (classes), slots (attributes), and pre-defined instances of concepts. Figure 3 shows the "is-a" (subclass) relationship between concepts as solid arrows. Slots and their cardinality restrictions are denoted with dashed lines. Instances are connected to their concepts by dotted lines. Fig. 3 contains some of the concepts needed to describe first-order theorem proving services and their results. One crucial concept (circled in Fig. 3) is the FO-ATP-Result which denotes results of first-order ATPs. A FO-ATP-Result can have a *time* slot which contains an instance of a time resource description (Time-Resource), and a *proof* of a conjecture (an instance of Proof). Most important, the *state* slot of a FO-ATP-Result always contains one of the valid states of first-order ATPs that we developed jointly with GEOFF SUTCLIFFE and STEPHAN SCHULZ [SZS03] for the MathWeb-SB. This state defines the prover's result for the

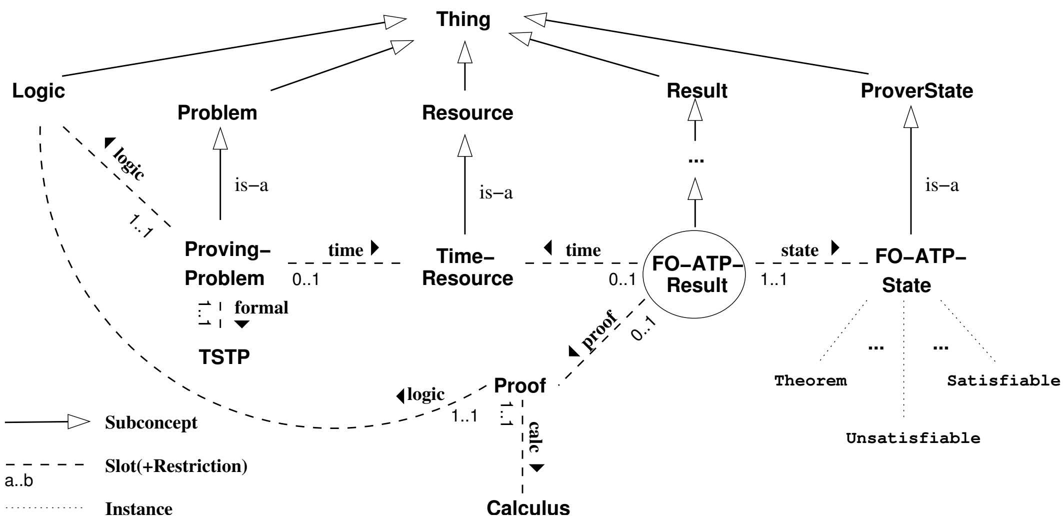

Figure 3: An ontology for first-order theorem provers.

conjecture given. For instance, the state `Theorem` says that the prover has found out that the given conjecture is a theorem.

3.1.2 Capability-based Selection and Combination of Mathematical Services

The second part of the project is concerned with the integration of problem solving capabilities into MathWeb-SB. The scenario can be sketched as follows: MathWeb-SB is given a mathematical problem or task formulated in the description language and ontology as defined in the first part of the project. The problem solver uses the declarative descriptions of the capabilities of the integrated mathematical services in order to come up with an adequate selection and combination of those services, that can potentially solve that problem.

A necessary prerequisite for the design of a procedure that searches for a combination is the definition of the primitives of a combination language. An example for such a primitive is sequential composition $S; S'$ of services, where S' tackles the subtask that remains after S. Another example is a divide-and-conquer approach that divides a conjunctive task into independent subtasks. This is expressed by parallel composition of services $S \parallel S'$, where S and S' both solve a subtask.

Based on the combination language we want to design and integrate a search procedure into MathWeb-SB. The primitives essentially impose a partial ordering on the combined services. This corresponds to the structure of AI plans, which are composed of partially ordered plan operators. Thus, the key idea is to wrap a service inside of a plan operator that describes its capabilities, and to use AI planning techniques as the search procedure. The resulting partially ordered plan is then straightforwardly transformed into an adequate combination.

Up to now we distinguish three different types of services:

Proving and Computing services are services that actually solve given problems: Examples are theorem provers, that try to prove a given formula, and computer algebra systems that simplify mathematical problems or solve polynomials.

Transformation and translation services are used to change the representation of a problem or a result. Examples are the transformation of some first-order logic problem into a propositional representation, or into a constraint problem, or into a mathematical problem that is in the domain of some computer algebra system.

Categorisation services are services that come up with a refined classification of a given problem. Examples are services that recognize that a problem is essentially a propositional problem, or that it belongs to the guarded fragment, or that it is a constraint problem over real numbers, or other subclasses that can be tackled by special decision procedures. The categorisation services may also recognize that a problem consists of different subproblems, each falling into a different class. The inherent problem modeling these services as plan operators is that although the precondition can be formulated in a declarative way, the actual postconditions depend on the computation of the involved categorisation service. This problem needs to be taken into account for the design of the AI planning procedure.

These different types of services influence the actual planning procedure, as some of the planning operators are useless unless its corresponding service is executed. This is typically the case for the categorisation services. The plan operators for the other services contain sufficient declarative descriptions that allow for planning without actually executing the service. The benefit of avoiding the execution of a service is that the time required by such a service is saved at planning time. Only when a plan is actually found are the services executed.

Since some services need to be executed during plan formation, this requires an interleaving of plan formation and plan execution, which is a well-known technique in AI planning. This requires usually a linearization of some part of a partially ordered plan, in order to allow for its execution. This is, however, certainly not adequate for our purpose, as the partial order of our plan represents exactly the sequential and parallel composition of services. But fortunately a linearization of the plan is not necessary in our case, since only intermediate steps need to be executed, namely those plan steps that involve categorisation services. This is a new style of interleaved plan formation and plan execution, which needs to be taken into account for the design of the planning procedure.

Once a plan is found, there are several ways to communicate the results: First, not every plan is completely executable, since it may involve services which are only references to techniques described in a publication. In those cases the result returned by the broker must contain both the results obtained for the executable parts as well as the non-executable parts of the plan. Secondly, even for fully executable plans, there are different possibilities to return a result, depending on what kind of result the requesting program or user is interested in. For example, the result may simply be the result obtained at the end of plan execution, i.e. the results returned by the last services of the plan. Alternatively the sequence of intermediate results or even the plan itself can be useful information, e.g. when using MathWeb-SB as a problem solver. In that case the sequence of steps and intermediate results provide some evidence how the given problem was solved. In the final phase of the project we will therefore develop modalities to communicate the solution, ranging from pure results computed by the used services, via the description of a combination of the used services, up to a heterogenous proof that provides evidence about the validity of the result.

3.1.3 An Example for Advanced Brokering

Given a subproblem of a large proof development undertaken with a higher-order theorem proving environment like ΩMEGA (based on a variant of simple type theory) or NuPrl (based on constructive type theory) the user may want to call one or more theorem provers available within a mathematical service network to tackle the problem. More specifically, let us assume that the subproblem consists of a set of (local) proof assumptions Γ and a conjecture ψ. Then the user or the proof assistant may pose the following two queries to a MathWeb broker:

- **Query 1:** Given Γ and ψ, determine if ψ is a logical consequence of Γ. We denote this query with the sequent $\Gamma \overset{?}{\vdash}_{HO} \psi$, where the subscript HO means that we deal with some variant of a higher-order logic.
- **Query 2:** Given Γ and ψ, find a natural-deduction proof object for ψ using Γ. We denote this query with the sequent $\Gamma \overset{P?}{\vdash}_{HO} \psi$, where P? expresses that the user is actually asking for an explicit proof object.

Assuming now that formal descriptions of the following mathematical services are available to the MathWeb broker:

HO2FO: Transforms higher-order problems (e.g., in type theory) into first-order logic if possible (such a service is implemented in the ΩMEGA system).

FO2CNF: Transforms first-order problems into their clause normal form (CNF) including Skolemization (there are already several different implementations of efficient CNF transformations available).

FOP2ND: Transforms a first-order resolution proof into a natural deduction proof. This service is, for instance, offered by the TRAMP system [Mei00] (available in the MathWeb-SB).

HOP2ND: Transforms a higher-order resolution proof into a natural deduction proof.

HO-ATP: Calls a higher-order theorem prover such as LEO (resolution approach) or TPS (matrix search). They take a problem description as an input and deliver an ATP-Result, i.e. either a valid state or a proof object.

FO-ATP: Calls a classical resolution-based first-order theorem prover which tries to find a proof for a given conjecture, and returns also an ATP-Result.

Dec-Proc: A decision procedure for a decidable fragment of first-order logic (e.g., Presburger Arithmetic or some guarded fragment). It takes the CNF of a first-order problem and returns a valid decision procedure state (DP-State), i.e. either valid or unsatisfiable.

The challenge consists in providing a brokering mechanism that plans the selection and suitable combination of logic-transformation-services, proving services, and proof transformation services in order deliver finally a customized solution. Figure 4 shows all possible sequences of services that may answer **Query 1** in a disjunctive tree. Probably, the problem can be solved directly by one of the available higher-order theorem provers. It may however essentially be a first-order problem and is therefore transformable using HO2FO. After this transformation, the problem can then be sent to the available first-order ATPs or it may be translated into clause-normal-form (FO2CNF) which can then be used as input for a decision procedure.

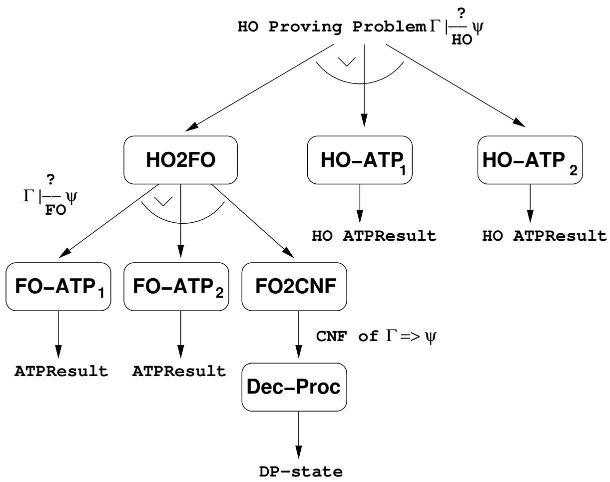

Figure 4: Possible sequences of service calls to answer Query 1.

In the case of **Query 2**, a proof object in a natural deduction calculus has to be delivered. Therefore, most decision procedures are no longer admissible because, in general, they only determine if an input formula is valid or unsatisfiable. Figure 5 shows possible sequences of services that may deliver the desired proof object. The most important aspect here is the transformation service HOP2ND which transforms a HO resolution-proof into a Natural Deduction proof. The disjunctive plans shown in Figure 4 and 5 could also be used to model parallel service invocations either to obtain a second, independent result for a problem or to increase the overall performance. Similar scenarios could be sketched with respect to the combination of decision procedures or symbolic computation engines. Or alternatively, a customized combination of problem re-representation services, and (one or more) decision procedures may be able to successfully answer a service request.

These scenarios show our approach to model semantical brokering *as a planning problem* based on the service descriptions of the service systems. This view of semantical brokering is well in accord with the new proof planning paradigm, which was invented and largely established by Alan Bundy (University of Edinburgh) and the applicants of this project.

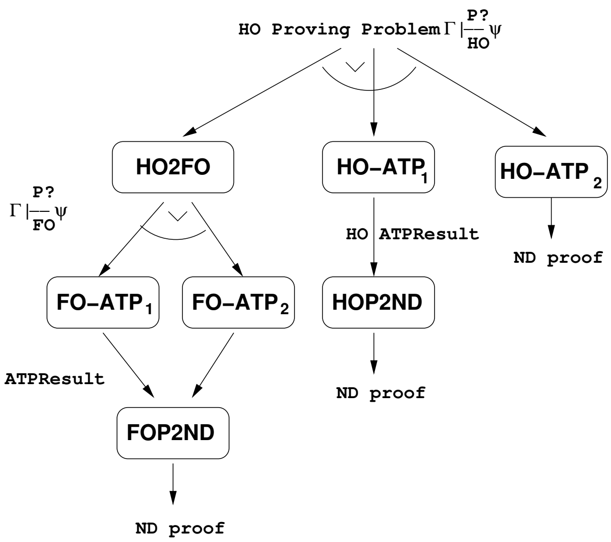

Figure 5: Possible sequences of service calls for Query 2.

3.2 Work Schedule (Arbeitsprogramm)

Task 1 Development of a service description language in close cooperation with MONET and MathBroker project. (IU Bremen)

Task 1.1 Definition and implementation of a description language for mathematical services based on existing XML standards.

Task 1.2 Ontology engineering for the language developed in Task 1.1.

Task 1.3 Description of existing services in the MathWeb-SB using the language and ontology developed in Task 1.1 and 1.2.

Task 2 Intelligent brokering of mathematical services. (Saarland Univ., IU Bremen)

Task 2.1 Translation of service descriptions into planning operators.

Task 2.2 Development of a planner for intelligent service brokering

Task 2.3 Case studies in combining heterogeneous mathematical services with focus on deduction systems.

Task 3 Maintenance and adaptation of current MathWeb-SB. (Saarland Univ.)

Task 3.1 Support for current MathWeb-SB users named in section 2.2.

Task 3.2 Porting parts of the current MathWeb-SB according to the results of the research made in Task 1 and Task 2.

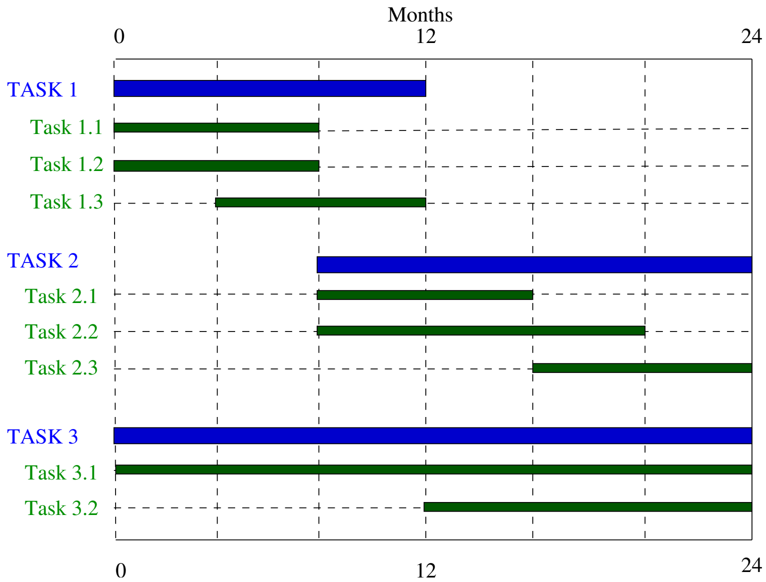

Figure 6: Scheduling of project tasks over 24 months.

4 Funds requested (Beantragte Mittel)

4.1 Staff (Personalbedarf)

1. *Two* researchers BAT IIa (1 Saarland University, 1 International University Bremen) for the complete application period.

2. *Four* research assistants (Hilfskräfte, 2 Saarland University, 2 International University Bremen, jeweils eine als wissenschaftliche und eine als studentische Hilfskraft) for 19 hours per week for the complete application period.

In Saarbrücken we plan to employ Dipl.-Inform Jürgen Zimmer. In 2000, Mr. Zimmer completed his Diploma Thesis in the Saarbrücken group which was about the combination of proof planning with constraint solving techniques. Since then he was responsible for the development of the MathWeb-SB system and participated in writing this project proposal. As a Young Visiting Researcher of the EU Training Network CALCULEMUS he independently worked on the integration of client-services at the University of Genova, Italy, and Edinburgh, Scotland. Hence, the main developer of MathWeb-SB, the largest network of mathematical service to-date, will be available for the realization of the project.

The high need of research assistants is due to the large amount though necessary implementational work to be realized in the project. As the implementation work will need a high level academic maturity, we ask that two of the research assistants be salaried as "wissenschaftliche Hilfkraft", so that we can also take candidates with a Diplom or Master's degree into consideration.

4.2 Scientific equipment (Wissenschaftliche Geräte) .... None

4.3 Consumables (Verbrauchsmaterial) .... None

4.4 Travel expenses (Reisen)

2000 Euro per annum (1000 Euro/year for Saarland Univ., 1000 Euro/year for IU Bremen)

4.5 Publication costs (Publikationskosten) .... None

4.6 Other costs (Sonstige Kosten) .... None

5 Preconditions for carrying out the project (Voraussetzungen für die Durchführung des Vorhabens)

5.1 Our team (Zusammensetzung der Arbeitsgruppe)

5.1.1 Saarland University

- Prof. Dr. J. Siekmann (C4, Univ. des Saarlandes)
- Dr. S. Autexier (Deutsches Forschungszentrum für Künstliche Intelligenz)
- Dr. C. Benzmüller (C2, Univ. des Saarlandes)
- Dr. D. Hutter (Deutsches Forschungszentrum für Künstliche Intelligenz)
- Dr. A. Meier (Deutsches Forschungszentrum für Künstliche Intelligenz)
- Dr. E. Melis (Deutsches Forschungszentrum für Künstliche Intelligenz)
- Dr. C.-P. Wirth (Bat IIa, Univ. des Saarlandes)

5.1.2 International University Bremen

- Prof. Dr. M. Kohlhase
- Dipl.-Phys. Immanuel Normann

5.2 Co-operation with other scientists (Zusammenarbeit mit anderen Wissenschaftlern)

At Saarland University we will cooperate with the OMEGA project funded in the SFB 378 *Resource-adaptive Cognitive Processes* (Prof. Siekmann), to which this work is complementary. At DFKI we will cooperate with the VSE group (Dieter Hutter and Werner Stephan) and the ACTIVEMATH-group (Erica Melis).

5.3 Foreign contacts and co-operations (Arbeiten im Ausland und Kooperation mit ausländischen Partnern)

The VERMATH project will be linked to the CALCULEMUS and MKM research initiatives. The aim is to provide an infrastructure that spins the web of mathematical services developed by the partners and supports their problem oriented selection and combination.

One of the applicants is the coordinator of the EU Research Training Network CALCULEMUS and of the successor proposal CALCULEMUS-II. Both partners are active nodes in the European MKM research initiative.

More specifically, research collaboration is planned with the following foreign partners:

- Tasks 1 & 2: The group of Manfred Kerber and Volker Sorge (University of Birmingham, UK).
- Task 1: Members of the MONET-project, the *MathBroker*-project, and the Mowgli-project, e.g. Prof. B. Buchberger (RISC Linz, Austria), Prof. Th. Hardin (Univ. Paris VI, France), Prof. A. Asperti (Univ. Bologna, Italy).
- Task 2 & 3: The Dream group of Prof. A. Bundy (University of Edinburgh, Scotland).
- Task 2 & 3: Members of the LOGOSPHERE-project[6], e.g. Prof. F. Pfenning and Prof. P. Andrews (CMU, Pittsburgh, USA), Prof. C. Schürmann (Yale University, New Haven USA), N. Shankar (SRI, Menlo Park, USA).

5.4 Scientific equipment available (Apparative Ausstattung)

The scientific equipment will be covered by the *Grundausstattungen* of Prof. J. Siekmann at Saarland University and Prof. M. Kohlhase at IU Bremen.

5.5 General contribution (Laufende Mittel für Sachausgaben)

- The yearly general contribution of consumables from the Saarland University via the *Grundausstattung* of Prof. J. Siekmann will be approx. 1000 Euro/year.

- The yearly general contribution of consumables from the IU Bremen via the *Grundausstattung* of Prof. M. Kohlhase will be approx. 1000 Euro/year.

5.6 Other requirements (Sonstige Voraussetzungen) .... None

6 Declarations (Erklärungen)

6.1

Ein Antrag auf Finanzierung dieses Vorhabens wurde bei keiner anderen Stelle eingereicht. Wenn wir einen solchen Antrag stellen, werden wir die Deutsche Forschungsgemeinschaft unverzüglich benachrichtigen.

[6] http://www.logosphere.org

6.2

- The *Vertrauensdozent* of the Saarland University, Prof. Dr. Janocha, has been informed about this research grant application.
- The International University Bremen is not yet a member university of DFG. Prof. Dr. Gerhard Haerendel, Dean of the School of Engineering and Science has been notified of this grant proposal.

6.3 Does not apply

7 Signatures (Unterschriften)

Dr. S. Autexier	Dr. C. Benzmüller
Prof. Dr. M. Kohlhase	Prof. Dr. J. Siekmann

8 Appendices (Anlagen)

(A) Curriculum vitae and list of publication Dr. S. Autexier

(B) Curriculum vitae and list of publication Dr. C. Benzmüller

(C) Curriculum vitae and list of publication Prof. Dr. J. Siekmann

(D) Curriculum vitae and list of publication Prof. Dr. M. Kohlhase

References

[ACE$^+$00] Stuart F. Allen, Robert L. Constable, Rich Eaton, Christoph Kreitz, and Lori Lorigo. The Nuprl open logical environment. In McAllester [McA00], pages 170–176.

[Ben03] Christoph Benzmüller. The CALCULEMUS research training network: A short overview. In *Proceedings of the First QPQ Workshop on Deductive Software Components at CADE-19*, pages 13–27, Miami, USA, 2003.

[BG00] D. Brickley and R.V. Guha. The Resource Description Framework. W3C Recommendation 1.0, World Wide Web Consortium, 2000. Available at http://www.w3.org/RDF/.

[CBE+92] J. Carbonell, J. Blythe, O. Etzioni, Y. Gil, R. Joseph, D. Kahn, Craig. Knoblock, S. Minton, M. A. Pérez, S. Reilly, M. Veloso, and X. Wang. PRODIGY 4.0: The Manual and Tutorial. CMU Technical Report CMU-CS-92-150, Carnegie Mellon University, june 1992.

[CC98] Olga Caprotti and Arjeh M. Cohen. Draft of the Open Math standard. The Open Math Society, `http://www.nag.co.uk/projects/OpenMath/omstd/`, 1998.

[CCMW01] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Service Description Language. W3C Recommendation 1.1, World Wide Web Consortium, 2001. Available at http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

[CIMP01] David Carlisle, Patrick Ion, Robert Miner, and Nico Poppelier. Mathematical Markup Language (MathML) version 2.0. W3c recommendation, World Wide Web Consortium, 2001. Available at http://www.w3.org/TR/MathML2.

[Con02] The MONET Consortium. The MONET project. http://monet.nag.co.uk/cocoon/monet/index.html, April 2002.

[DCN+00] Louise A. Dennis, Graham Collins, Michael Norrish, Richard Boulton, Konrad Slind, Graham Robinson, Mike Gordon, and Tom Melham. The prosper toolkit. In *Proceedings of the Sixth International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS-2000*, LNCS, Berlin, Germany, 2000. Springer Verlag.

[EHN94] K. Erol, J. Hendler, and D. Nau. Semantics for hierarchical task network planning. Technical Report CS-TR-3239, UMIACS-TR-94-31, Computer Science Department, University of Maryland, March 1994.

[FHLW03] Dieter Fensel, James A. Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003. ISBN 0-262-06232-1.

[Fie01] Armin Fiedler. Dialog-driven adaptation of explanations of proofs. In Bernhard Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1295–1300, Seattle, WA, 2001. Morgan Kaufmann.

[Fip02] Fipa. The foundation for intelligent physical agents, specifications. http://www.fipa.org, 1998-2002.

[GM93] M. J. C. Gordon and T. F. Melham. *Introduction to HOL – A theorem proving environment for higher order logic*. Cambridge University Press, 1993.

[Gua91] N. Guarino. A concise presentation of itl. In H. Boley and M. M. Richter (Eds.), editors, *Processing Declarative Knowledge*, pages 199–220. Springer Verlag, 1991.

[HNC+03] Jason Hickey, Aleksey Nogin, Robert L. Constable, Brian E. Aydemir, Eli Barzilay, Yegor Bryukhov, Richard Eaton, Adam Granicz, Alexei Kopylov, Christoph Kreitz, Vladimir N. Krupski, Lori Lorigo, Stephan Schmitt, Carl Witty, and Xin Yu. METAPRL: A modular logical environment. In David Basin and Burkhart Wolff, editors, *Proceedings of the 16th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2003)*, volume 2758 of *LNCS*, pages 287–303, Rome, Italy, 2003. Springer.

[HvHPS01] Ian Horrocks, F. van Harmelen, and P. Patel-Schneider. DAML+OIL, March 2001. Available at http://www.daml.org/2001/03/daml+oil-index.

[Koh01] Michael Kohlhase. OMDoc: Towards an internet standard for the administration, distribution and teaching of mathematical knowledge. In Eugenio Roanes Lozano, editor, *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2000*, number 1930 in LNAI. Springer Verlag, 2001.

[Mar01] Jonathan Marsh. Xml base. W3c recommendation, The World Wide Web Consortium, 2001. available at `http://www.w3.org/TR/xmlbase/`.

[McA00] David McAllester, editor. *Automated Deduction – CADE-17*, number 1831 in LNAI. Springer Verlag, 2000.

[McC90] William McCune. Otter 2.0. In Mark Stickel, editor, *Proceedings of the 10th Conference on Automated Deduction*, number 449 in LNCS, pages 663–664, Kaiserslautern, Germany, 1990. Springer Verlag.

[Mei00] Andreas Meier. System description: TRAMP: Transformation of machine-found proofs into ND-proofs at the assertion level. In McAllester [McA00], pages 460–464.

[ORS92] S. Owre, J. M. Rushby, and N. Shankar. PVS: a prototype verification system. In D. Kapur, editor, *Proceedings of the 11th Conference on Automated Deduction*, volume 607 of *LNCS*, pages 748–752, Saratoga Spings, NY, USA, 1992. Springer Verlag.

[Red96] Darren Redfern. *The Maple Handbook: Maple V Release 4*. Springer Verlag, Berlin, 1996.

[RV01] Alexander Riazanov and Andrei Voronkov. Vampire 1.1 (system description). In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Proceedings of the First International Joint Conference on Automated Reasoning (IJCAR 2001)*, volume 2083 of *LNAI*, pages 376–380, Siena, Italy, June 2001. Springer.

[SBB+02] J. Siekmann, C. Benzmüller, V. Brezhnev, L. Cheikhrouhou, A. Fiedler, A. Franke, H. Horacek, M. Kohlhase, A. Meier, E. Melis, M. Moschner, I. Normann, M. Pollet, V. Sorge, C. Ullrich, C.P. Wirth, and J. Zimmer. Proof development with ΩMEGA. In A. Voronkov, editor, *Proceedings of the 18th Conference on Automated Deduction (CADE–18)*, volume 2392 of *LNAI*, pages 143–149, Copenhagen, Denmark, 2002. Springer Verlag, Germany.

[SC01] W. Schreiner and O. Caprotti. The MathBroker Project. http://poseidon.risc.uni-linz.ac.at:8080/index.html, October 2001.

[SD02] Michael Sintek and Stefan Decker. Triple—a query, inference, and transformation language for the semantic web. In *Proc. of the International Semantic Web Conference (ISWC)*, Sardinia, June 2002.

[SMVW02] M.K. Smith, D. McGuinness, R. Volz, and C. Welty. Web ontology language, November 2002. Available at http://www.w3.org/TR/2002/WD-owl-guide-20021104/.

[SWKL01] K. Sycara, S. Widoff, M. Klusch, and J. Lu. LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Journal on Autonomous Agents and Multi-Agent Systems, Kluwer Academic*, 4(4), 2001.

[SZS03] Geoff Sutcliffe, Jürgen Zimmer, and Stephan Schulz. Communication fomalisms for automated theorem proving tools. In Volker Sorge, Simon Colton, Michael Fisher, and Jeremy Gow, editors, *Proceedings of the Workshop on Agents and Automated Reasoning, 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

[Tea] Coq Development Team. *The Coq Proof Assistant Reference Manual*. INRIA. see `http://coq.inria.fr/doc/main.html`.

[Wei97] Christoph Weidenbach. SPASS: Version 0.49. *Journal of Automated Reasoning*, 18(2):247–252, 1997. Special Issue on the CADE-13 Automated Theorem Proving System Competition.

[Wic99] Gerhard Jürgen Wickler. *Using Expressive and Flexible Action Representations to Reason about Capabilities for Intelligent Agent Cooperation*. PhD thesis, University of Edinburgh, 1999.

[Wol99] Stephen Wolfram. *The Mathematica book (5th edition)*. Cambridge University Press, 1999.

[ZK02] Jürgen Zimmer and Michael Kohlhase. System description: The MathWeb software bus for distributed mathematical reasoning. In Andrei Voronkov, editor, *18th Conference on Automated Deduction (CADE-18)*, volume 2392 of *LNAI*, Copenhagen, Denmark, 2002. Springer.