System Description: The MathWeb Software Bus for Distributed Mathematical Reasoning

Jürgen Zimmer[1] and Michael Kohlhase[2]

[1]FB Informatik, Universität des Saarlandes
jzimmer@mathweb.org
[2]School of Computer Science, Carnegie Mellon University,
kohlhase+@cs.cmu.edu

1 Introduction

Automated reasoning systems have reached a high degree of maturity in the last decade. Many reasoning tasks can be delegated to an automated theorem prover (ATP) by encoding them into its interface logic, simply calling the system and waiting for a proof, which will arrive in less than a second in most cases. Despite this seemingly ideal situation, ATPs are seldom actually used by people other than their own developers. The reasons for this seem to be that it is difficult for practitioners of other fields to find information about theorem prover software, to decide which system is best suited for the problem at hand, installing it, and coping with the often idiosyncratic concrete input syntax. Of course, not only potential outside users face these problems, so that, more often than not, existing reasoning procedures are re-implemented instead of re-used.

In a larger context, the same problems surface in many application domains, and have led to the emerging field of "web-services", where (parts of) applications are provided as so-called services on the Internet. The main problem is in providing a standardized infrastructure for identifying, discovering and calling these web services. Even though the field has got a lot of (industrial) attention because of potential applications in e-commerce, development and deployment are still in their early stages.

In [FK99], we have already reported on the MathWeb-SB service infrastructure specialized on deduction and symbolic computation services. The system MathWeb-SB[1] connects a wide-range of reasoning systems (*mathematical services*), such as ATPs, (semi-)automated proof assistants, Computer Algebra Systems (CASs), model generators (MGs), constraint solvers (CSs), human interaction units, and automated concept formation systems, by a common *mathematical software bus*. Reasoning systems integrated in the MathWeb-SB can therefore offer new services to the pool of services, and can in turn use all services offered by other systems.

[1] We used to call the MathWeb-SB simply MathWeb in earlier publications. The latter is now used for a whole set of projects cooperating to provide an infrastructure for web-based and semi-automated mathematics (see `http://www.mathweb.org`).

This system description summarizes the development in the last three years. We further extended the list of reasoning systems integrated in the MathWeb-SB, stabilized existing integrations and explored new application domains for the MathWeb-SB (see section 2). The main improvements are a more flexible architecture (section 3), and increased standards support in the communication protocols used in MathWeb-SB (section 4). As a consequence, it is much simpler now to use and integrate mathematical services into the MathWeb-SB infrastructure.

2 New Features and Reasoning Systems

The MathWeb-SB now offers an efficient version of the tptp2X utility [SSY94] that allows to transform first order problems in TPTP format (CNF of FOF) into the input format of most existing ATPs. With our tptp2X service (based on a servlet) we could reduce the transformation time from 3-4 seconds (of the tptp2X shell command) down to 100-200 ms (of the servlet) for a transformation of a medium size TPTP problem. This speedup is crucial since typical client applications produce thousands of first order conjectures and send them to the MathWeb-SB. All first order ATPs in the MathWeb-SB, such as *Bliksem*, E, Otter, Spass, and Vampire, now accept problems in TPTP format which they translate into their native format using the tptp2X service.

The HR program [Col00] performs automated theory formation in different domains. During concept formation, HR produces up to 3700 first order conjectures per minute. We integrated HR as a server and as a client into the MathWeb-SB. HR can now send its conjectures in TPTP format to single provers or to the concurrent ATP service. The latter allows to run several provers in parallel on one or several problems. An application using concurrent ATP, like HR, can specify whether it is interested in the first result returned by a prover or in all results to compare the success or the runtime of the ATPs.

Together with A. Adams, we integrated the specification and verification system PVS [ORS92] in the MathWeb-SB and defined a special `provePVS` service that uses the automated proof procedure of PVS. The `provePVS` service accepts conjectures in OpenMath and PVS syntax. Using the `provePVS` service with the *transcendental library* of PVS [Got00], we proved conjectures about the continuity of real-valued functions in the Ωmega proof planner.

3 Architectural Improvements

Since [FK99], we have further modularized and stabilized the MathWeb-SB and have built a stable network of reasoning systems that is in everyday use at different Universities in Europe and the US (cf. Fig. 1, which we will use as a concrete example for our discussion).

While the first version of the MathWeb-SB was based on one central broker (a facilitator service that allows clients to discover services), it is now based on a dynamic net of brokers. Brokers maintain a database of local services offered by *meta-services* and can be given a set of URLs that refer to *remote brokers*, i.e. to other brokers available in the Internet. During startup, a broker tries to connect to his remote brokers and to inform them about his address in the Internet and about its local services. This registration mechanism significantly improved the stability and availability of the MathWeb-SB: even if one or more brokers become unavailable at the same time (e.g. due to machine- or network failure), there are still some others left that may offer the lost services.

Service requests by client applications, e.g. by the ΩMEGA proof assistant [SB02], are forwarded to all known remote brokers if the requested service is not available locally. This allows, e.g., an ΩMEGA client running at the University of Birmingham to use Computer Algebra Systems offered to a broker at the University of Saarbrücken.

Fig. 1. The MathWeb Software Bus

4 Interfaces

At a conceptual level, MathWeb-SB now offers a uniform interface to all first order ATPs, abstracting away from system peculiarities. The main idea is that a generic `prove` service exports the state of the prover after processing a problem description given in a standard format (e.g., TPTP, DFG, or Otter). A call to the system is modeled as a request to the service to achieve a certain state (e.g. one, where a proof has been found). We have extended the specification of ATP states proposed in [AKR00] by states which describe errors, timeouts and situations where the search is exhausted for some reason. We extended all first order ATP services in the MathWeb-SB such that the `prove` service always returns one of the valid ATP states.

On a the protocol level, MathWeb-SB has been extended to include a native http interface and – building on that – an XML-RPC [XRP] interface. As a consequence MathWeb-SB services can, for instance, be accessed via HTML forms. We have chosen XML-RPC (an XML encoding of Remote Procedure Calls) to be the standard protocol for external access to MathWeb-SB since it is an independent standard that is easy to understand and has been implemented in many different programming languages (there are currently 56 independent implementations available). XML-RPC allows client applications outside the MathWeb-SB to request service objects and to use their service methods[2]. For instance, a client can simply request a service object for the ATP SPASS by sending the XML content in Fig. 2 via an http POST request to a MathWeb-SB XML-RPC server. An XML-RPC implementation in the client's implementation language

```
<methodCall><methodName>Broker.getService</methodName>
 <params><param><value><string>SPASS</string></value></param></params>
</methodCall>

<methodCall><methodName>prove</methodName>
 <params><param><struct>
  <member><name>1</name><value><string>
   include('Axioms/EQU001+0.ax').
   include('Axioms/GRP004+0.ax').
   input_formula(conjecture118,conjecture,(! [B,C,D] :
    ((equal(inverse(B),C) &amp; equal(multiply(C,B),D) ) &lt;=>
      (equal(multiply(B,C),D) &amp; equal(multiply(C,B),D) &amp; equal(inverse(C),B))))).
   </string></value></member>
   <member><name>syntax</name><value><string>tptp</string></value></member>
   <member><name>timeout</name><value><int>40</int></value></member>
  </struct></param></params>
</methodCall>
```

Fig. 2. Discovering SPASS and sending it a problem with XML-RPCs.

simplifies this process drastically since it abstracts from http protocol details and offers XML-RPCs using data structures of the host language. As a consequence, developing MathWeb clients is quite simple in such languages. Last but not least, both MS Internet Explorer and the open source WWW browser Mozilla now allows to perform XML-RPC calls within JavaScript. This opens new opportunities for building user interfaces based on web browsers.

5 Conclusion, Availability and Future Work

We have presented new developments in the MathWeb-SB system, a framework for web-services specialized to deduction and symbolic computation services. The main new developments in the last three years have been a more flexible architecture and the support of standardized communication protocols.

[2] More information about service access via XML-RPC can be found at http://www.mathweb.org/mathweb/xmlrpc/howto.html.

The MathWeb-SB has been implemented in the concurrent constraint programming language MOZART [Moz] which is now available as Version 1.2.3 for many platforms (Unix, Linux, MacOS X, and Windows). An easy to install binary distribution of the compiled MathWeb-SB code and further information is available at `http://www.mathweb.org/mathweb/`. The system sources can be obtained via anonymous CVS under the GNU General Public License.

The MathWeb-SB is currently used in many automated reasoning groups around Europe and the US. Among other applications, it supports the use of external reasoning systems, such as CASs, ATPs, and MGs, in the ΩMEGA proof planner, as well as the use of MAPLE exercises in the web-based learning system *ActiveMath* [Mel00]. It offers an efficient access to state-of-the-art first order ATPs including an efficient transformation between different problem formats. This is crucial for applications that produce many first order conjectures, like, for instance, the HR system.

The next development steps in MathWeb-SB will be further support of standards (e.g. the emerging SOAP standard), and further agentification based on "service descriptions". These are machine-understandable specifications of the reasoning and computation capabilities of the web-services that can lead to service discovery and system-independent service requests. Our uniform interface to the first-order theorem provers in MathWeb-SB is a first step into this direction, but a lot of conceptual work remains to be done for more complex services, such as constraint solvers, decision procedures, or symbolic computation systems.

References

[AKR00] A. Armando, M. Kohlhase, and S. Ranise. Communication protocols for mathematical services based on KQML and OMRS. In *Proc. of the Calculemus Symposium 2000)*, St. Andrews (Scotland), August 6–7, 2000.

[Col00] S. Colton. *Automated Theory Formation in Pure Mathematics.* PhD thesis, University of Edinburgh, Edinburgh, Scotland, 2000.

[XRP] XML Remote Procedure Call Specification. `http://www.xmlrpc.com/`.

[Moz] The mozart programming system.`http://www.mozart-oz.org`.

[FK99] A. Franke and M. Kohlhase. System description: MATHWEB, an agent-based communication layer for distributed automated theorem proving. In H. Ganzinger, ed., *Proc. CADE-16*, *LNAI* 1632, pp. 217–221. Springer 1999.

[Got00] H. Gottliebsen. Transcendental Functions and Continuity Checking in PVS. In Proc. of TPHOLs'00, *LNCS* 1869, pp. 197–214, Springer 2000.

[Mel00] E. Melis. The 'Interactive Textbook' project. In D. McAllester, ed., *Proc. of CADE WS "Deduction and Education;"*, *LNAI* 1831. Springer Verlag, 2000.

[ORS92] S. Owre, J. M. Rushby, and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, ed.*Proc. of CADE-11*, *LNCS* 607, pp. 748–752. Springer 1992.

[SB02] J. Siekmann, C. Benzmüller et al. Proof development with ΩMEGA. In A. Voronkov, ed., *Proc. of CADE-18*, *LNAI* Springer 2002.

[SSY94] G. Sutcliffe, C. Suttner, and T. Yemenis. The TPTP problem library. In A. Bundy, ed., *Proc. of CADE-12*, *LNAI* 814, pp. 252–266. Springer 1994.