The MathServe System
for Semantic Web Reasoning Services

Jürgen Zimmer\textsuperscript{1} and Serge Autexier\textsuperscript{1,2}

\textsuperscript{1} FB Informatik, Universität des Saarlandes, D-66041 Saarbrücken, Germany
\textsuperscript{2} DFKI, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
\{jzimmer, autexier\}@ags.uni-sb.de

1 Introduction

In recent years, formal verification of hardware and software components has increasingly attracted interest from both academia and industry. The widespread use of automated reasoning techniques requires tools that are easy to use and support standardised protocols and data exchange formats. In [1] the first author presented the MathWeb Software Bus, a first step towards re-usable reasoning services. The MathWeb-SB had several drawbacks which limited its usability. For example, it had no service brokering capabilities and the user had to know exactly which reasoning system to use to solve a problem and how to access it.

Here we present the MathServe system that overcomes the limitations of the MathWeb-SB. MathServe offers reasoning systems as Semantic Web Services described in OWL-S [2]. MathServe’s service broker can automatically find suitable services (and compositions of services) to solve a given reasoning problem. The use of Semantic Web technology allows applications and humans to automatically retrieve and access reasoning services via the Internet. MathServe complements similar projects, such as the MathBroker project [3], which describe computational services as Semantic Web Services. We provide an overview of the MathServe system in §2 and describe the evaluation of the system at CASC-20 in §3. We conclude and discuss future work in §4.

2 The MathServe System

The MathServe system is based on state-of-the-art technologies for Semantic Web Services: It integrates reasoning systems as Web Services. The semantics of these Web Services is described in the OWL-S upper ontology for Web Services [2]. OWL-S service profiles define the inputs and outputs as well as the preconditions and effects of Web Services. MathServe services and the service broker are accessed by means of standard Web Service languages and protocols.

Client applications can interact with MathServe in two principal ways: All reasoning services can be invoked individually with reasoning problems. Complex queries can be sent to the MathServe broker which can perform service
matchmaking and automated service composition. Given a query containing a reasoning problem, service matchmaking returns a list of standalone services that can potentially answer that query. So far, service matchmaking simply performs class subsumption tests on the types of input and output parameters of available services and the query provided. If no single service can answer a query, MathServe’s service composer can automatically combine services using classical AI planning and decision-theoretic reasoning.

Reasoning problems and their solutions are encoded in OWL/RDF format [4]. The primary interface to MathServe is the standard Web Service interface defined in the Web Service Description Language (WSDL). Services are invoked via the Simple Object Access Protocol (SOAP). Tools and libraries for WSDL and SOAP are available in many mainstream programming languages. Next to the SOAP interface, the MathServe broker offers an XML-RPC interface with convenient interface methods similar to the ones described in [1].

**Reasoning Services in MathServe.** MathServe provides several reasoning systems for classical first-order logic with equality as Semantic Web Services: 1) Services for clause normal form transformation of first-order problems are provided by the tptp2X utility and the FLOTTER system (available with SPASS [5]). 2) A problem analysing service can determine the Specialist Problem Class of a theorem proving problem (see below). 3) Deduction services are provided by state-of-the-art Automated Theorem Proving (ATP) systems. 4) Transformations of formal proofs are performed by the systems Otterfier [6] and TRAMP [7]. We cannot describe all these services in this paper. Detailed descriptions of the services provided by Otterfier and TRAMP can be found in [6]. In this paper, we focus on first-order ATP services. With respect to ATP services, MathServe is similar to the SSCPA system [1], which has been developed for human users, while MathServe’s services can be consumed by software applications.

The latest version of MathServe offers the ATP systems DCTP 10.21p, EP 0.91, Otter 3.3, SPASS 2.2, Vampire 8.0 and Waldmeister 704 as Semantic Web Services (see [5] for ATP system descriptions). All theorem proving services support the same Web Service interface and can be invoked with a theorem proving problem, a CPU time limit, and (optionally) prover-specific options. The answers provided by ATP services specify unambiguously what has been established by the underlying system. For this, we developed an ontology of 18 well-defined ATP statuses [8]. Furthermore, results of ATP services contain the complete output of the prover, a reference to the problem submitted, the CPU and wall-clock time used, and, if available, resolution proofs in the new TPTP format.

**OWL-S Descriptions of ATP Services.** The performance of an ATP system depends on the computational resources given to the prover as well as the type of the problem to solve. Sutcliffe and Suttner [9] identified six “objective problem features” that have an impact on the performance of ATP systems. The

---

1 Accessible via the TPTP web site [http://www.tptp.org](http://www.tptp.org).