System Description: MBase, an Open Mathematical Knowledge Base

Andreas Franke and Michael Kohlhase

FB Informatik, Universität des Saarlandes
afranke|kohlhase@ags.uni-sb.de

Abstract. In this paper we describe the MBase system, a web-based, distributed mathematical knowledge base. This system is a mathematical service in MathWeb that offers a universal repository of formalized mathematics where the formal representation allows semantics-based retrieval of distributed mathematical facts.

1 Introduction

Around 1994, an anonymous (but well-known) group of authors put forward the "Qed Manifesto" [QED95], which advocates building up a mathematical knowledge base (and supporting software systems) as a kind of "human genome project" for the deduction community. Unfortunately, the vision has failed to catch on in spite of a wave of initial interest. In our view this is largely due to the lack of supporting software, as well as to the ensuing debate on the "right" logical formalism.

In this paper we describe the MBase system, a web-based mathematical knowledge base (see `http://www.mathweb.org/mbase`). It offers a the infrastructure for a universal, distributed repository of formalized mathematics. Since it is independent of a particular deduction system and particular logic[1], the MBase system can be seen as an attempt to revive the Qed initiative from an infrastructure viewpoint. The system is realized as a mathematical service in the MathWeb system [FK99], an agent-based implementation of a mathematical software bus for distributed theorem proving.

We will start with a description of the system from the implementation point of view in the next section (we have described the data model and logical issues in [KF00]). In section 3, we will take a brief look at the interface protocols based on the OpenMath and Kqml standards (see [FHJ+99,Koh00]). This reliance of Internet standards for communication makes MBase an open system, and the implementation presented in this paper just one of its possible instances.

2 Architecture and Implementation

The MBase system is realized as a distributed set of MBase servers (see figure 1). Each MBase server consists of a Relational Data Base Management

[1] See [KF00] for the logical issues related to supporting multiple logical languages while keeping a consistent overall semantics.

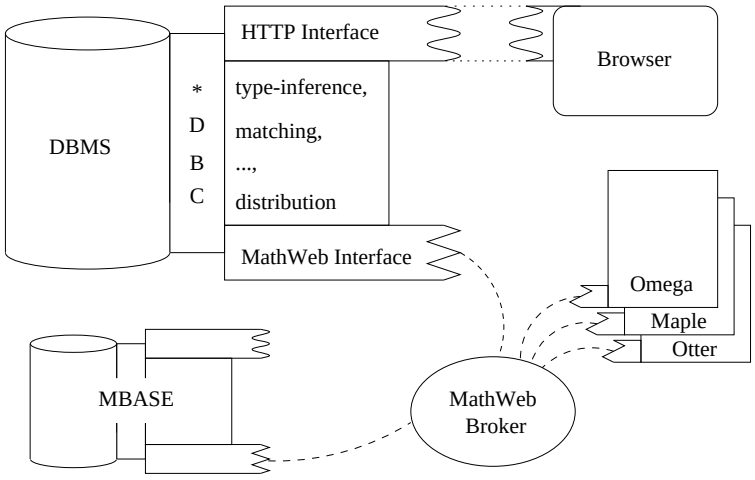

Fig. 1. System Architecture

System (RDBMS) e.g. ORACLE connected to a MOZART process (yielding a MATHWEB service) via a standard data base interface (in our case JDBC). For browsing the MBASE content, any MBASE server provides an `http` server (see `http://mbase.mathweb.org:8000` for an example) that dynamically generates presentations based on HTML or XML forms.

This architecture combines the storage facilities of the RDBMS with the flexibility of the concurrent, logic-based programming language OZ [Smo95], of which MOZART is a distributed implementation (see `http://www.mozart-oz.org`). Most importantly for MBASE, MOZART offers a mechanism called **pickling**, which allows for a limited form of persistence: MOZART objects can be efficiently transformed into a so-called pickled form, which is a binary representation of the (possibly cyclic) data structure. This can be stored in a byte-string and efficiently read by the MOZART application effectively restoring the object. This feature makes it possible to represent complex objects (e.g. logical formulae) as OZ data structures, manipulate them in the MOZART engine, but at the same time store them as strings in the RDBMS. Moreover, the availability of "OZlets" (MOZART functors) gives MBASE great flexibility, since the functionality of MBASE can be enhanced at run-time by loading remote functors. For instance complex data base queries can be compiled by a specialized MBASE client, sent (via the Internet) to the MBASE server and applied to the local data e.g. for specialized searching (see [Duc98] for a related system and the origin of this idea).

MBASE supports transparent distribution of data among several MBASE servers (see [KF00] for details). In particular, an object O residing on an MBASE server S can refer to (or depend on) an object O' residing on a server S'; a query to O that needs information about O' will be delegated to a suitable query to the server S'. We distinguish two kinds of MBASE servers depending on the data

they contain: *archive servers* contain data that is referred to by other MBases, and *scratch-pad* MBases that are not referred to. To facilitate caching protocols, MBase forces archive servers to be *conservative*, i.e. only such changes to the data are allowed, that the induced change on the corresponding logical theory is a conservative extension. This requirement is not a grave restriction: in this model errors are corrected by creating new theories (with similar presentations) shadowing the erroneous ones. Note that this restriction does not apply to the non-logical data, such as presentation or description information, or to scratchpad MBases making them ideal repositories for private development of mathematical theories, which can be submitted and moved to archive MBases once they have stabilized.

3 Interface Language

The primary interface language of MBase is the Xml-based markup language OMDoc [Koh00], a document-centered extension of the emerging OpenMath standard [CC98] for mathematical objects. For instance the definition of a double function would be of the following form.

```
<definition id="double.def" item="double.sym" type="simple">
 <CMP xml:lang="eng">The doubling function defined by addition</CMP>
 <FMP><OMOBJ><OMBIND>
  <OMS cd="stlc" name="lambda"/>
  <OMBVAR><OMV name="X"/></OMBVAR>
  <OMA><OMS cd="arith1" name="plus"/><OMV name="X"/><OMV name="X"/></OMA>
 </OMBIND></OMOBJ></FMP>
</definition>
```

The `CMP` (commented mathematical property) element gives an informal characterization of the definition (which is a simple definition for the symbol with the identifier `double.sym` according to the attributes to the `definition` element) and the `FMP` (formal MP) gives the defining λ-term $\lambda X.(+XX)$ in OpenMath representation. Note that the question of the semantics of such a term is determined by that of the symbols λ and $+$. These are specified in the MBase theories given in the `cd` attributes of the `OMS` elements (the name of the symbol together with the theory establish unique reference in MBase)

As a consequence of the Xml-based approach it is possible to generate other logical formats from OMDoc by specifying simple Xsl [Dea99] style sheets; in fact the transformation from OMDoc to the input formats of the Ωmega [BCF+97] and InKa [HS96] theorem provers is realized this way. It should be an easy exercise for most other concrete input formats. Furthermore one can generate customized OMDoc documents from MBase, which can then be presented in one of the more standard presentation media (e.g. LaTeX or HTML/MathML).

Generating OMDoc from a reasoning system is also quite simple in practice, since OMDoc has a relatively simple structure (fully specified in an Xml document type definition [Koh00]) that closely follows the term structure of OpenMath

(using the `OMS`, `OMV`, `OMA`, `OMBIND` elements to describe formula trees made up of symbols, variables, applications and abstractions).

4 Conclusion, Evaluation and Future Work

We have described the MBase system, a distributed mathematical knowledge base, it can be obtained from `http://www.mathweb.org/mbase`. This system differs from other repositories of mathematical data such as the Isabelle [Isa] or Pvs [PVS] libraries in that it is an independent system not tied to a particular deduction system and offers inference services (matching, type-computation,...). The data format is not geared towards a particular application.

It is currently used by the ΩMEGA and InKa theorem provers for storing and sharing logical theories including theorems, definitions, tactics and methods. In particular, the MBase service can be used as an ontology server fixing the semantics of mathematical objects used in protocols for deduction system integration. Furthermore, the MBase system is used as the basis of an interactive personalized mathematics book (IDA [CCS99]). Here, the structure information contained in the MBase version of the IDA data can be used to generate individualized sub-documents of IDA on the fly. While in the first case study the logical formulation of mathematical data is in the center of interest, in the second application textual representation plays a much more prominent role. MBase supports both formats and even fosters their integration.

The current implementation uses the very simple file-based `gdbm` database system. This is sufficient for the amount of data currently available in ΩMEGA, InKa and IDA. Furthermore it offers a very flexible, open and portable programming base. A version of MBase that uses Oracle is currently under development.

Here a comparison to the MDB system [Har97] developed at the University of Erlangen is in order. MDB aims at supplying database support for the Mizar libraries, and is based on an object-oriented extension of Oracle. Unfortunately, already the first 13 (of more than 300) articles already need 500 MB disc space in Oracle. Our division of labor that treats logical formulae in the programming language mozart and relational, text and structural data in a DBMS pays off here. The size of the data base is only one order of magnitude larger than the size of the OMDoc encoding, which is comparable in size to the encodings used e.g. in ΩMEGA, Isabelle, or Pvs. As an example for relative sizes of representations in MBase we consider the core theory library of ΩMEGA and the IDA text:

Relative sizes of representations in MBase (MB)			
System	native	OMDoc	MBase
ΩMEGA	0.61 (POST)	1.5	4.2
IDA	4.2 (LaTeX)	5.0	9.3

Even when MBase implementations based on industrial strength relational data base systems like e.g. Oracle are available, we believe that the current `gdbm`-based implementation can still serve as a local development knowledge base and

"proxy" system to ease the load on the central MBase repository servers. Such a local system will probably also be better suited to support the operations necessary for changing definitions and axiomatizations during the development of a theory.

In the current version, we have not yet treated more advanced structuring concepts like theory morphisms, inheritance wrt. signature mappings, etc. that have been developed for structuring the knowledge base (see [KF00]). There remains much to be done in this direction, and we hope to adopt techniques from algebraic specification (see for instance [Hut99]).

References

[BCF+97] C. Benzmüller, L. Cheikhrouhou, D. Fehrer, A. Fiedler, X. Huang, M. Kerber, M. Kohlhase, K. Konrad, E. Melis, A. Meier, W. Schaarschmidt, J. Siekmann, and V. Sorge. ΩMEGA: Towards a mathematical assistant. In William McCune, editor, CADE'97, Springer LNAI 1249, pages 252–255, 1997.

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

[CCS99] Arjeh Cohen, Hans Cuypers, and Hans Sterk. *Algebra Interactive!* Springer, 1999. Interactive Book on CD.

[Duc98] Denys Duchier. The NEGRA tree bank. Private communication, 1998.

[FHJ+99] Andreas Franke, Stephan M. Hess, Christoph G. Jung, Michael Kohlhase, and Volker Sorge. Agent-oriented integration of distributed mathematical services. *Journal of Universal Computer Science*, 5:156–187, 1999.

[FK99] Andreas Franke and Michael Kohlhase. System description: MathWeb, an agent-based communication layer for distributed automated theorem proving. In H. Ganzinger, editor, CADE'99, Springer LNAI 1632, pages 217–221, 1999.

[Har97] Michael Hartmeier. Aufbau einer Datenbank f"ur mathematisches Wissen. Master Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 1997.

[HS96] Dieter Hutter and Claus Sengler. INKA - The Next Generation. In M.A. McRobbie and J.K. Slaney, editors, CADE'96, Springer LNAI 1104, pages 288–292, 1996.

[Hut99] Dieter Hutter. Reasoning about theories. Technical report, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), 1999.

[Isa] The isabelle online theory library. Internet interface at `http://www4.informatik.tu-muenchen.de/~isabelle/library-Isabelle98-1`.

[KF00] Michael Kohlhase and Andreas Franke. Mbase: Representing knowledge and context for the integration of mathematical software systems. *Journal of Symbolic Comutation*, 2000. forthcoming.

[Koh00] Michael Kohlhase. OMDoc: Towards an OPENMATH representation of mathematical documents. Seki Report SR-00-02, Fachbereich Informatik, Universität des Saarlandes, 2000. `http://www.mathweb.org/ilo/omdoc`.

[PVS] The Pvs libraries. `http://pvs.csl.sri.com/libraries.html`.

[QED95] The QED manifesto. Internet Report `http://www.cybercom.net/~rbjones/rbjpub/logic/quedres00.htm`, 1995.

[Smo95] G. Smolka. The Oz programming model. In Jan van Leeuwen, editor, *Computer Science Today*, Volume 1000 of *LNCS*, pages 324–343. Springer, 1995.

[Dea99] Stephen DeachExtensible stylesheet language (xsl) specification. W3c working draft, W3C, 1999. Available at `http://www.w3.org/TR/WD-xsl`.