MathDox: Mathematical Documents on the Web
Contribution to the OMDoc book

A.M. Cohen, H. Cuypers, E. Reinaldo Barreiro
Department of Mathematics
Eindhoven University of Technology

December 2, 2005

Abstract

This section deals with our work on interactive mathematical documents that make use of the World Wide Web. These documents take input from various sources, users, and mathematical services. Communication between these different entities can be realized using OpenMath. But, such communication and the interactivity inside the mathematical document take place in a specific, dynamic, context. In this paper we discuss our approach to such a dynamic mathematical context: MathDox. It consists of both an XML based markup language for interactive mathematical contents and a set of software tools realising the interactivity.

1 Introduction

Although the notion of an interactive mathematical document has been around for several years, cf. [5], its realization is nowhere near the final stage. Recent Web technological progress, for instance, has enabled a much smoother communication of mathematics than ever before. The use of an interactive mathematical document (IMD) can provide a window to the world of mathematical services on the internet, and a mathematical service on the internet can be created by the building of an interactive mathematical document. The purpose of this section is to describe MathDox, an ensemble of software tools for creating IMDs. It is a rather complex software system including

1. an XML based language that offers markup support for the source texts of IMDs;

2. a document server, rendering interactive mathematical documents from source text and interactively obtained information;

3. mathematical services, providing connections with CASs like Mathematica and GAP via OpenMath phrasebooks (cf.[10]).

The creation of MathDox is a project at the Technische Universiteit Eindhoven (the RIACA institute). Several people at RIACA have helped creating it; here we mention Manfred Riem, Olga Caprotti, Hans Sterk, Henny Wilbrink, Mark Spanbroek, Dorina Jibetean, ...

The system is mainly built with Java and related technology.

2 The Language

The MathDox source is an XML document. We have derived our own XML schema for these source texts. We have been influenced by both DocBook [3] and OMDoc [9]. The former is a fairly general standard for electronic books, the latter is a very rich, and strongly logic-oriented standard for mathematical documents—the main subject of this book. There is a close link with OMDoc. Both OMDoc and MathDox use OpenMath [10], the difference being that OMDoc focuses on formalizing mathematics whereas MathDox focuses on interactivity. The connections with both DocBook and OMDoc are of importance to us because we expect several authoring tools for it to emerge in the coming few years, and we want to profit from their presence.

The mathematics in the MathDox source is given by means of OpenMath objects. This feature has clear advantages in terms of portability. The DocBook type grammar sees to it that there are natural scopes, where mathematical objects 'live'. For instance, when a chapter begins with "Let $\mathbb{F}$ be a field", the scope of the variable $\mathbb{F}$ is assumed to be the whole chapter (although, somewhere further down the hierarchy, say in a section of the chapter, this assignment can be overridden). Within the MathDox grammar, special attention is also given to interactivity.

Interactivity in MathDox is taken care of by XML tags representing various programming constructs as well as queries to external mathematical services. These actions take place within part of the context, which fixes the precise sematics of the objects involved. Further constructs are available for handling context and user input. Our notion of context is based on [7]. Context is divided into static and dynamic context. The static context may be defined as the set of all XML sources from which a interactive document can be prepared for use. Two extreme forms are OpenMath Content Dictionaries and a chapter of an ordinary book. The dynamic context behaves more like the state of a CAS. It keeps track of the variables introduced, their properties, their values, and their scopes. The language has constructs for storing and changing this information. For example, the field $\mathbb{F}$ introduced at the beginning of a chapter may be specified to a finite field of five elements in the context of a particular section of the chapter.

Although semantics is the primary target, some features for presentation have been built into the language. In order to have a flexible presentation, we use presentation annotated OpenMath. In MathDox we allow style attributes inside OpenMath objects. By discarding these style attributes, regular OpenMath is obtained. For instance, by providing the appropriate value for the style attribute, the author has a choice between a slash and a fraction display. In

$$\frac{3/4 + 2/3}{5}$$

we have used them both.

Another way of solving presentation issues is illustrated by the statement: $3, 4 \in \mathbb{Z}$. The corresponding OpenMath expression would be the equivalent of $3 \in \mathbb{Z} \wedge 4 \in \mathbb{Z}$, but our OpenMath statement reads that the sequence $3, 4$ belongs to $\mathbb{Z}$. So here, the semantics of the element-of symbol has been stretched so as to help out presentation.

3 The Software

An essential component of the MathDox software is its document server. It provides a view to the client of the content and realizes both the static and the dynamic context. The usage of the MathDox document server is shown in Figure 1. We explain in some detail the main components shown in this picture.

1. The *client*. The client is represented by a math-enabled Web browser. It will present views of the serviced documents to the user, interact with the user, and communicate user input to the document server.

The communication between client and server takes place via the HTTP request/response mechanism. The responsibility for interaction is mostly on the server side.

2. The *document server*. This server caters for presentation, communication, and context. It supports a wide range of actions ranging from handling

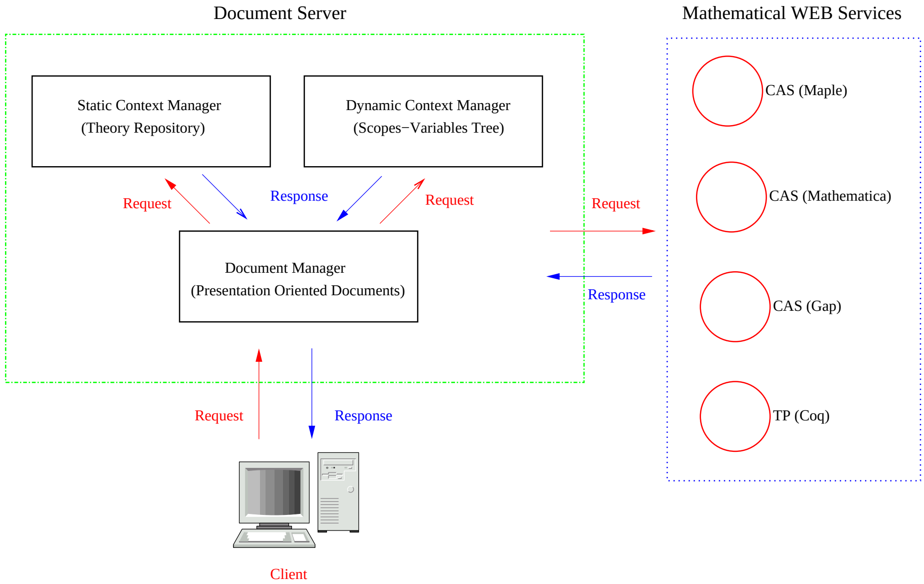

Figure 1: The MathDox software

queries to searching within documents for mathematical content and from placing (and retrieving) objects into the context, to rendering documents in different views.

The document server is realized as a Java enhanced Web application [6] inside a Web server. It is not a monolithic entity. As shown in Figure 1, it is formed by different parts. The *document manager* serves views to the client. IMDs can be thought of as programs (scripts) encoding the production of a response. In generating the response, they can make use of the information contained in the static context, and in the dynamic context (scopes and variables), the user input communicated along with request, and the results of computations carried on by one or more mathematical services.

Another part is the *static context manager* which is responsible for managing a repository of MathDox mathematical theories.

The final (third) part is the *dynamic context manager* which is responsible for the dynamic information.

3. *Mathematical services.* Mathematical services can be very diverse: some may serve as general interfaces to CAS or to Theorem Provers. The MathDox software provides ways to access these services via standard protocols, among which those developed under the MONET project [8]. The mecha-

nism extends the phrasebook set-up for OpenMath, [4, 1]. For constructing specific OpenMath services, we employ our Java OpenMath library ROML [11].

4 Conclusion

Now that MathDox is close to a complete working version, trial applications are in the make. We mention

- a server for providing designs of experiments on command to statisticians,
- an exercise repository for the EU funded LeActiveMath project,
- a mathematics course on calculus, with automated natural language production from a formal-mathematical source for the EU funded project WebALT,
- interactive lecture notes (the successor of [2]) for an Abstract Algebra course within a mathematically oriented Bachelor curriculum,
- educational material for highschool mathematics in the Netherlands.

References

[1] O. Caprotti, A. M. Cohen, and M. Riem. *Java Phrasebooks for Computer Algebra and Automated Deduction.* SIGSAM Bulletin, 2000. Special Issue on OpenMath.

[2] A.M. Cohen, H. Cuypers, H. Sterk. *Algebra Interactive!*, Interactive lecture notes on Algebra (paper book and CD-Rom), Springer-Verlag, Heidelberg, August 1999.

[3] DocBook, `http://www.docbook.org`.

[4] Olga Caprotti, Arjeh M. Cohen, Hans Cuypers, Manfred N. Riem, and Hans Sterk. *Using OpenMath Servers for Distributing Mathematical Computations,* pp. 325–336 in: ATCM 2000: Proceedings of the Fifth Asian Technology Conference in Mathematics, Chiang-Mai, Thailand, Wei Chi Yang, Sung-Chi Chu, Jen-Chung Chuan (eds.), ATCM, Inc., 2000.

[5] A.M. Cohen and L. Meertens. *The ACELA project: Aims and Plans,* pp. 7–23 in *Computer-Human interaction in Symbolic Computation* (ed. N. Kajler), Texts and Monographs in Symbolic Computation, Springer-Verlag, Wien, 1998

[6] JavaServer Pages, for dynamically generated Web content, `java.sun.com/products/jsp/`.

[7] A. Franke and S. Hess and Ch. Jung and M. Kohlhase and V. Sorge, Agent-Oriented Integration of Distributed Mathematical Services, Journal of Universal Computer Science, vol 5, (1999) 156–187.

[8] MONET, an EU funded project, `www.monet.nag.co.uk`.

[9] OMDoc, a standard for open mathematical documents, `www.mathweb.org/omdoc/`.

[10] OpenMath Society Website, `www.openmath.org`.

[11] ROML, The RIACA OpenMath Library, `crystal.win.tue.nl/download/`.