MathBook: web technology for interactive mathematical documents

Hans Cuypers   Hans Sterk

Research Institute for Applications of Computer Algebra (RIACA)
Department of Mathematics and Computer Science
Eindhoven University of Technology, The Netherlands
{hansc,sterk}@win.tue.nl

1 Introduction

Internet technology is opening up new possibilities for constructing and structuring documents. This paper deals with the efforts of the RIACA group at Eindhoven University of Technology to help shape a new generation of interactive documents, particularly in the field of mathematics. In creating such a new generation of interactive mathematical documents, our research combines web technology with our experience in mathematics teaching. It should be emphasized that for our purposes the web technological side is far from trivial, even though a glance at the internet may easily lead to the impression that this fancy looking world leaves nothing to desire. The fact that web technology development is not primarily driven by mathematical or teaching interests, is a good indication that the combination of mathematics, teaching and web technology forms a field of research that is still open for further exploration.

In this paper we outline an architecture we have implemented to realize our ideas so far. We discuss the functionalities that have been included as well as their mathematical motivation, and we focus on teaching aspects. Technical details will be ommitted and discussed elsewhere. We also outline the future plans of RIACA. Our work is a continuation of the research that has led to *Algebra Interactive!* [3], teaching material which we use at Eindhoven University of Technology for two undergraduate algebra courses.

2 Web technology learning environment

Starting 1998, all students entering Eindhoven University of Technology acquire a laptop (on reasonable terms). Through their laptops students have access to the internet at various places on campus, from cantines to lecture rooms. Laptops provide students with a digital environment, that we want to integrate with our basic algebra courses. Since algebra is often regarded as a dull subject, we took this lack of popularity as an extra motivation to enliven algebra courses and show its usefulness and ubiquity in the sciences.

The main feature of web technology that makes it so useful, is its potential to extend one's digital environment to web services all over the world. Surfing to static pieces of information on the internet is the most commonly known web activity, but web services of a more sophisticated nature are envisioned for our interactive documents.

Let us first briefly outline how *Algebra Interactive!* came into being.

2.1 Algebra Interactive!

When dealing with mathematics and mathematics teaching, various aspects immediately come to mind: mathematical expressions or formulas (for example $(x+1)^3 - \sin(x)$), computations (both numerical and symbolical computations), vizualizations (graphs, schematic illustrations, simulations), the logical structure of a body of theory (what is needed to explain a concept like prime number?), exercises (open and multiple choice). Our first aim in *Algebra Interactive!* [3] was to accomodate for all these aspects, with the technology then available (1997) or within our reach with some additional developmental research.

Algebra Interactive! is interactive course material on elementary university algebra, covering topics like basic arithmetic, modular arithmetic, polynomial arithmetic, permutations, monoids, groups, rings, fields and permutation groups. The main material of *Algebra Interactive!* comes on a CD-rom together with a book containing the main text. The interactive document is written in (dynamic) HTML and can be viewed through a standard web browser like Netscape 4.x or Internet Explorer 5.x. Mathematical formulas and symbols are displayed using images (GIFs). The document contains document specific navigation tools in addition to the standard navigation buttons. It has a layered structure, viz. is constructed in chapters, sections, pages and subpages. Pages represent the core information, but this

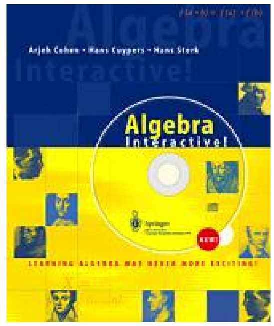


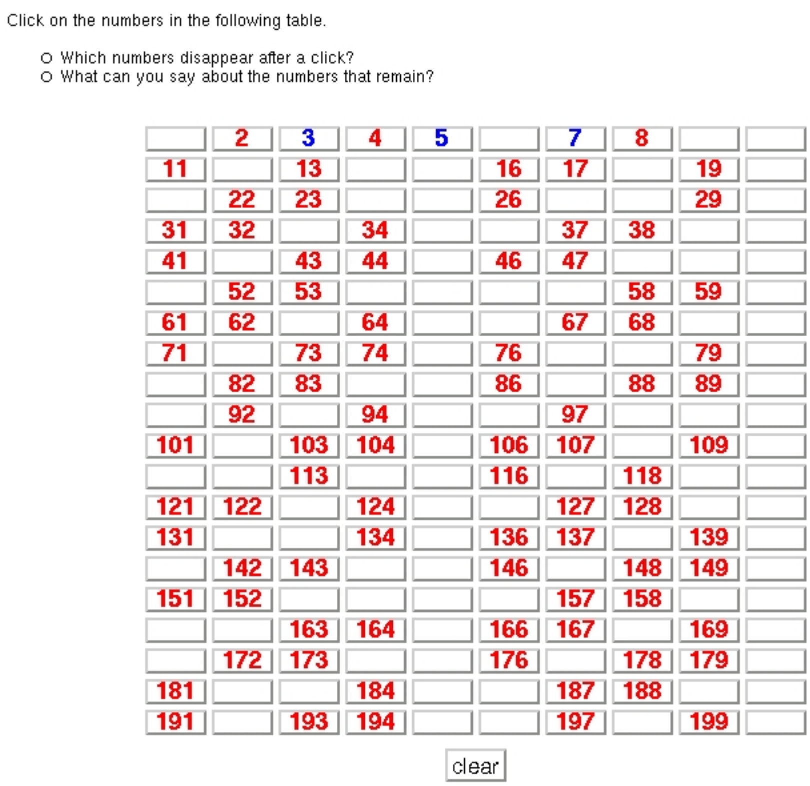
Figure 1: Algebra Interactive! Book and CD-rom.

information is restricted to the very heart of the matter, discussed and presented in one or two 'tableaux' per page with at most brief comments or a brief connecting text. The reader is invited to dig deeper via special buttons. For instance, proofs of theorems are not presented on these main pages, but are hidden under 'proof buttons', or: each tableau comes with a multiple choice question to test one's immediate grasp of the material. Many dynamical and interactive illustrations are added in the form of JAVA applets. These also include specialized calculators for each coherent set of computational techniques, like modular arithmetic or the calculus of permutations. At many places the computations require more specialized software. To provide such a tool, throughout the document so-called 'gapplets' are inserted, illustrations of a computational nature in input/output form, that enable users to interface with the computer algebra package GAP [4] (the word gapplet is a contraction of GAP and applet), without any specific knowledge of GAP. This enables students to concentrate on the mathematics involved. This connection with a 'back-engine' turned out to be one of the main technical obstacles in the development of *Algebra Interactive!*. Yet, providing a mathematical service of this kind was at the heart of our project as we will explain in the sequel.

Finally, apart from open questions (with online hints and solutions), the document contains a data base of multiple choice questions. The student is given the choice to do a multiple choice test on a chapter of his/her choice or all chapters up to a given point. A test is then randomly generated from the relevant pool of questions and, after completion, automatically evaluated.

2.2 Algebra Interactive! in the class room

Algebra Interactive! is used in our elementary algebra courses at Eindhoven University of Technology for first and second year mathematics and computer science students starting the academic year 1999-2000. Mathematics students attend 3 hour weekly sessions where theory, explanation, exercises and regular tests are integrated. Instructors usually use the dynamic parts of the material to start sessions with a motivating problem. For instance, the Sieve of Eratosthenes can serve as an introduction to prime numbers, the multiplicative building blocks of the integers. The corresponding illustration produces the integers $1, \ldots, 200$ on the screen.



	2	3	4	5		7	8		
11		13			16	17		19	
	22	23			26			29	
31	32		34			37	38		
41		43	44		46	47			
	52	53					58	59	
61	62		64			67	68		
71		73	74		76			79	
	82	83			86		88	89	
	92		94			97			
101		103	104		106	107		109	
		113			116		118		
121	122		124			127	128		
131			134		136	137		139	
	142	143			146		148	149	
151	152					157	158		
		163	164		166	167		169	
	172	173			176		178	179	
181			184			187	188		
191		193	194			197		199	

Figure 2: Erathostenes' Sieve.

Upon clicking on a given number (≥ 2), all its multiples except the number itself disappear. This leads to questions of the sort: What kind of numbers never vanish? Is there a systematic and efficient way of producing a list of prime numbers up to a given point? What is the complexity of such an algorithm? Previously discussed properties on division of integers then naturally lead to a discussion on prime numbers. Given the computational tools built into the document, the instructor can easily go beyond what is usually done on the blackboard, still emphasizing the conceptual background, and gently introducing these tools on the way.

Sessions may be interrupted for multiple choice questions to test the students' immediate grasp of a tableau containing a theorem or an algorithm, etc. Longer parts of the session are spent on open questions where students use or start using the available illustrations, computational tools (with no specific knowledge of the back engines needed) and hints.

For computer science students, the much larger number of students makes this set-up impossible. Instead, there is a separate course followed by practice sessions.

Experience over the last few years show a gradually increasing appreciation of the material on the side of the students; it seems as if they have to get used to work in a digital environment, whereas the instructors also have to get familiar with the new material. Students especially appreciate the multiple choice questions for testing their knowledge, and the specialized computational tools for helping them solve exercises.

2.3 Aims for a new edition, MathBook

At present it is clear that XML, the Extensible Markup Language [11], in combination with the programming language JAVA, is becoming the new standard for web documents. Its flexibility makes it extremely suitable for our purposes. RIACA's new interactive web documents and tools, for which we have adopted the name MATHBOOK, will be constructed in an XML/JAVA setting. MATHBOOK technology should extend the functionalities of *Algebra Interactive!* and introduce more dynamic and sophisticated ones along the lines we explain below. Our goal is twofold in the sense that we aim to develop tools to produce interactive mathematical documents in general and aim to produce a second edition of *Algebra Interactive!* in particular.

Separation of content and presentation

One of the main goals of the MATHBOOK project is to produce open, platform independent and semantically rich mathematical documents. In contrast to the presentation language HTML, used for our first version of *Algebra In-*

teractive!, the open and flexible structure of the meta-language XML makes it possible to realize this goal. By using XML in our MATHBOOK setting, it is possible to separate the content of our mathematical documents from the presentation. The mathematical content and structure of our documents are kept in XML-source files. By using XSL-stylesheets [12], we can transform these sources into several presentable formats ranging from LaTeX [7], the word processor especially adapted to the needs of the mathematics community, or PDF, for book quality print-outs, to HTML or JSP-pages [5] for highly dynamical and interactive web pages.

Not only can XSL-stylesheets be used to control the format of the presentation of the content, they can also be used to govern the contents of this presentation. In this way a presentation can vary from a printout of an overview of a chapter of *Algebra Interactive!* to a complete web version of the chapter including all kinds of dynamic examples.

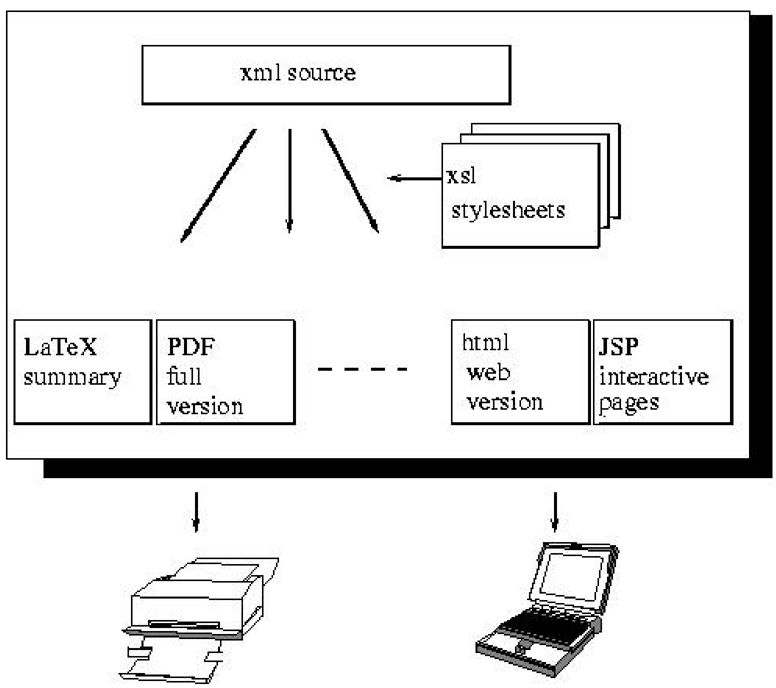


Figure 3: From source to presentation.

In the sequel we explain, mainly by examples, how we add structure and semantics to our sources, in which way we intend to add interactivity to our documents and which types of different presentations of the material we have in mind.

Meaningful mathematical expressions

The XML-source of a MATHBOOK document contains the structural information of the content. In particular, the material is divided into chapters, sections, pages, theorems, lemmas, examples, etc. Moreover, the relations between these various items is fixed inside this source, e.g., an example or an exercise to a theorem is linked to that theorem. The organisation of this structural information in the XML-source is pretty much along the lines of DOCBOOK [10], an XML-format for electronic books, and OMDOC [6], an XML-format mainly concerned with mathematical databases. Besides this structural information, we also want to capture the semantics of the material. In particular, we want to endow mathematical expressions with their meaning and provide for means to profit from such enriched objects. Usually, symbols (words, formulas, etc.) on the screen are interpreted by the user only. The machine is in no way aware of their mathematical content. For example, when we see $\pi(x + y)$ in the document, we may think, based on the context, that this refers to the product of the well-known constant π ($= 3.14159\ldots$) and $x + y$, but equally well it could be a projection, called π, applied to the sum of two vectors x and y. It would be a great step forward if the semantics of the mathematical object is stored in some way so that the meaning of the object can be retrieved and the object can in principle be (re)used for meaningful manipulations and computations. The OpenMath language [9] (also used in OMDOC) provides such a semantically rich representation of mathematics which facilitates the exchange of mathematics between software packages, including computer algebra systems like Mathematica, Maple or GAP. Since OpenMath is XML-compatible, we have adopted the OpenMath standard to incorporate mathematical objects into our XML-sources. Here is a simple example of an XML encoded OpenMath object representing $\pi(x + y)$, the product of the constant π with the sum of x and y.

```
<OMOBJ>
       <OMA><OMS cd="arith1" name="times"/>
            <OMS cd="nums1" name="pi"/>
            <OMA><OMS cd="arith1" name="plus"/>
                 <OMV name="x"/>
                 <OMV name="y"/>
            </OMA>
       </OMA>
</OMOBJ>
```

Notice that the product of π and $(x+y)$ is obtained by applying (OMA) the OpenMath symbol

```
 <OMS cd="arith1" name="times"/>
```

to the symbols

```
 <OMS cd="nums1" name="pi"/>
```

and

```
<OMA>
<OMS cd="arith1" name="plus"/>
    <OMV name="x"/>
    <OMV name="y"/>
</OMA>.
```

The meaning of the symbols is captured in so-called Content Dictionaries, which are specified in the "cd" attribute of the symbol.

Mathematical services

Through OpenMath we are able to exchange mathematical objects between various software packages, if needed over the net. In other words, requests regarding our mathematical objects can be handled by mathematical services locally or elsewhere, since our objects are endowed with their meaning. A mathematical expression in the document can be sent to a package, say Mathematica or Maple, for evaluation without specific knowledge of the software package, or its syntax in particular. Indeed, we have developped JAVA-tools that take care of the communication with various back-engines. Now, mathematical problems sometimes require the use of techniques from different areas: general purpose packages deal with a wide range of basic algorithms whereas specialized packages usually concentrate on algorithms applicable in a restricted field. Fortunately, the possibility of distributing mathematical computations is becoming a reality using internet technology, and in previous work [2] we have designed an architecture for distributing mathematical computations, a first step towards a framework where a range of mathematical services are available and fully integrated. A key issue here is that this usage of mathematical services is not restricted to the software available on the user's own machine: a computation can be dealt with by a mathematical service located somewhere else on the internet. Here is one aspect where the extension of one's digital environment is becoming a reality.

For teaching purposes OpenMath and the possibility of accessing mathematical services provide several advantages, ranging from fairly trivial to more sophisticated ones. For instance, the meaning of any mathematical expression in the document can be unambiguously retrieved. This is useful if doubt arises, say whether π is the name of a function, of a variable or denotes the famous constant. The use of computational back-engines enables one to illustrate theorems and the like. For example, not just a tangent to a smooth curve may be calculated, but with the help of the graphical power of packages like Mathematica or Maple it can also be visualized in a graph. Students can also experiment. For instance, to find a closed form for the function $f(n)$ defined by $f(n) := \Sigma_{i=1}^{n}\, i$. Also, automated testing of students by open questions becomes available. A student asked for an anti-derivative of $2 \cdot \sin x \cdot \cos x$ can give the correct answer $\sin^2 x$ but also $-\cos^2 x$ or even the maybe less obvious but still correct answer $\frac{\sin^2 x}{3} - \frac{2 \cdot \cos^2 x}{3}$. A mathematical back-engine can check the answer by taking derivatives. In short, bringing mathematical objects to life in the sense as described above, enables students to play around with them in a way comparable to the classical pen and paper situation where the student was in control of both meaning and manipulation. But the digital environment just sketched extends the scope of the student's actions enormously.

We have realized the above environment in the form of JSP-pages, which we derive from our XML-source, together with a set of JAVA-tools. A JSP-page is a web document in which there is a mixture of HTML or presentation-XML with (calls to) JAVA-code. The JAVA-tools govern the interaction with the mathematical services. They make it possible to contact these mathematical services, e.g., computational back-engines like Mathematica, Maple or GAP, and submit queries to them, but also take care of handling the responses obtained from these back-engines. For example, they take care of displaying a graph obtained from Mathematica or a table with the first 100 values of the function $f(n) := \Sigma_{i=1}^{n}\, i$ computed by GAP, or even praise the student for giving a correct answer to the question of finding an anti-derivative of $2 \cdot \sin x \cdot \cos x$.

Flexibility of content and presentation

This digital era requires a rethinking of the roles of electronic presentation and paper documents. The enormous success of paper documents over the past centuries is enough indication that abolishing the paper format is to be considered a premature act. It is more likely that in the near future a suitable combination of formats will be the standard. Finding a suitable balance is part of our quest.

As we already discussed before, our aim is to construct a single XML document that serves as basis for the generation of various formats, including formats suited for printing but also for dynamic web presentation. This source document should contain all the structural as well as semantic information, necessary to derive these various formats. The conversion to different formats, directed by XSL-transformations, is less trivial than it seems at first glance. A paper version for instance assumes a strictly linear ordering of the material, whereas electronic versions can be non-linear or even non-static. Before stating a short list of possibilities that have been implemented (in still rudimentary form; the tools to construct user friendly conversion options are still under development), let us elaborate a bit on the ways one may wish to structure documents for teaching purposes.

To begin with, *Algebra Interactive!* is used in courses for mathematics and computer science students, but the material is not specifically adapted to the needs of each of these groups. With MathBook technology we are able to generate from our source file several documents adapted to the needs of various groups of students. Here are a few examples you can think of. For instance, material for mathematics students could contain detailed proofs of statements, whereas material for computer science students only outlines proofs, and instead concentrates more on algorithmic aspects like complexity. An instructor's version including suggestions for presenting and dealing with the material can be generated. Or maybe you want to focus on a single topic from the material, say group theory. Then you would like to be able to generate from the source a coherent document dealing with groups only. Another feature is the possibility to produce documents, ranging from summaries to complete text books, and display them as interactive material on the web or print them on paper.

Here is a list of formats that we have so far implemented (although corresponding menu driven tools for authors are not yet available):

- Paper versions: from the XML source book quality printable formats can be generated via stylesheets that transform the document into LaTeX. This assumes that the XML source contains the relevant information to generate a complete, linearly ordered document.

- Linear electronic versions: linearly ordered documents including dynamic illustrations and computational facilities.
- Non-linear electronic versions: here, authors are presented with options to choose various ways of structuring and layering the document.

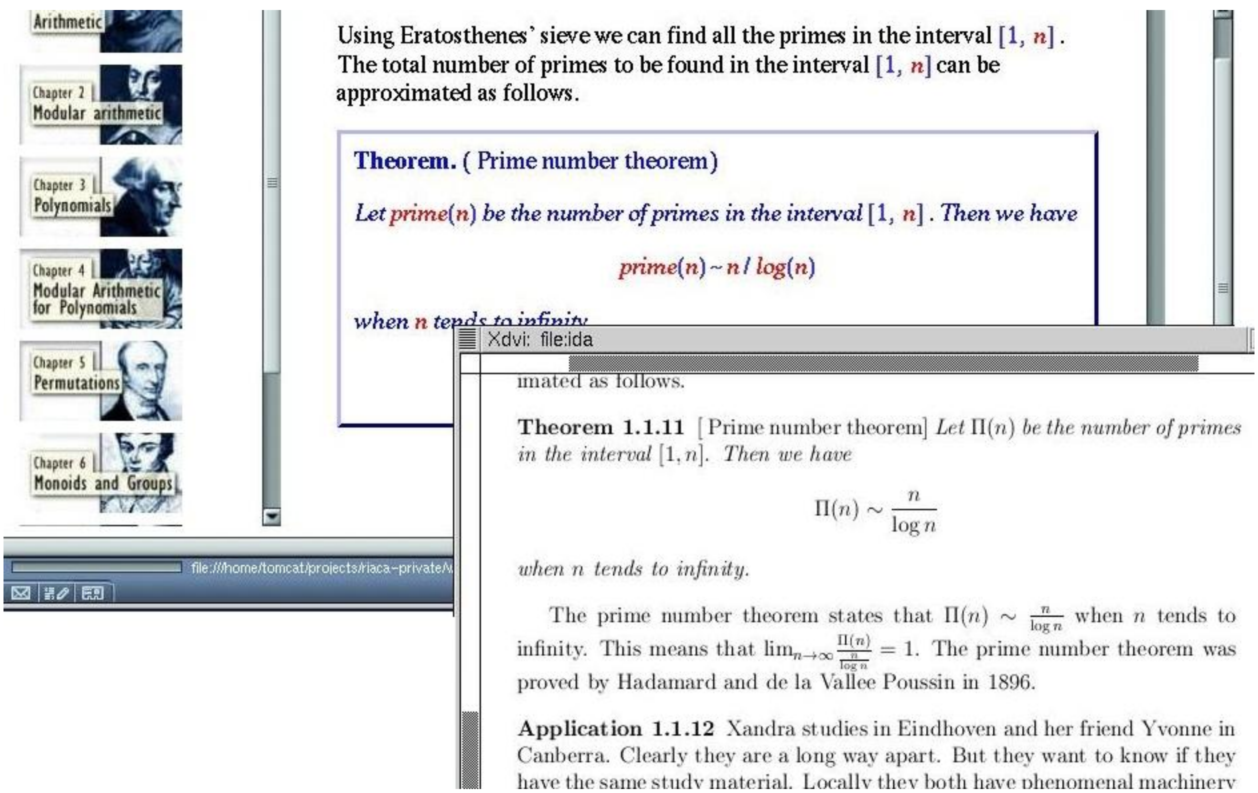


Figure 4: A web and printable version of a page.

Let us turn to the issue of presentation of mathematical symbols on the screen. Since OpenMath is compatible with MATHML [13], an XML compatible standard for presentation of mathematical expressions, presentation problems as with HTML are resolved as soon as MATHML is standardly supported by browsers (currently, we use the browser Mozzila [8] in our work since it supports MATHML). We no longer need to include mathematical expressions through gifs or similar cumbersome approaches.

Finally we should mention, that we are not yet planning to include adaptivity in the sense of for example De Bra et al. [1], where the document changes in response to the user's actions.

3 Conclusion

Our teaching experience with *Algebra Interactive!* and constant experimentation with emerging internet technologies in the RIACA group help shape our efforts to produce a new generation of interactive mathematical documents. We are nearing the end of the process of determining what such documents should look like in our view, given the present technological constraints. Parts of our view have been implemented and experimented with, but user friendly tools for potential authors are still to be developed. There are more technological obstacles still on our way. For instance, complete phrasebooks are not readily available and programming facilities are still under development to mention a few.

References

[1] P. De Bra, H. Wu, E. de Kort. *Design issues for general-purpose adaptive hypermedia systems*. Proceedings of the ACM Conference on hypertext and hypermedia, pp. 141-150, Aarhus, Denmark, 2001.

[2] O. Caprotti, A. M. Cohen, H. Cuypers, M. N. Riem and H. Sterk. *Using OpenMath servers for distributing mathematical computations*, ATCM 2000: Proceedings of the Fifth Asian Technology Conference in Mathematics, 2000, Chiang-Mai, Thailand, eds. Wei-Chi Yang and Sung-Chi Chu and Jen-Chung Chuan, 325–336.

[3] A. M. Cohen, H. Cuypers and H. Sterk. *Algebra Interactive!* Springer-Verlag, Heidelberg, Berlin etc., 1999.

[4] The GAP Group, *GAP – Groups, Algorithms, and Programming*, Version 4.2; 2000, `http://www.gap-system.org`.

[5] *Java Server Pages* JSP, `http://java.sun.com/products/jsp/`.

[6] M. Kohlhase, *OMDoc, Towards an Openmath Representation of Mathematical Documents*, DFKI, Saarbrücken, 1999, `http://www.mathweb.org/omdoc/`.

[7] Leslie Lamport, LaTeX, *A Document Preparation System*, Addison-Wesley, 1994.

[8] The Mozzila Organisation, `http://www.mozilla.org`.

[9] The OpenMath Consortium. The OpenMath Standard. OpenMath Deliverable 1.3.3a, OpenMath Esprit Consortium, `http://www.nag.co.uk/projects/OpenMath.html`, August 1999. O. Caprotti, D. P. Carlisle and A. M. Cohen Eds.

[10] Norman Walsh and Leonard Muellner, *DocBook: The Definitive Guide*, O'Reilly, 1999, `http://www.docbook.org`.

[11] World Wide Web Consortium, *Extensible Markup Language (*XML*)*, `http://www.w3.org/XML/`.

[12] World Wide Web Consortium, *Extensible Stylesheet Language (*XSL*)*, `http://www.w3.org/Style/XSL`.

[13] World Wide Web Consortium, *Mathematical Markup Language (*MATHML*)*, `http://www.w3.org/Math/`.