Formal Mathematics on the Web

Andrea Asperti, Luca Padovani, Claudio Sacerdoti Coen, Irene Schena
Department of Computer Science
Via di Mura Anteo Zamboni 7, 40127 Bologna, ITALY.
contact: asperti@cs.unibo.it

1 Introduction

The World Wide Web is already now the main repository and vehicle for mathematical knowledge, and its relevance is likely to be exponentiated by emerging publishing technologies like MathML[1]. However, almost all mathematical documents available on the Web are marked up only for presentation: they are machine readable, but not machine-understandable. This step is precisely the goal of the so called Semantic Web (see [6]): establishing a layer of machine understandable data will allow automated agents, sophisticated search engines and interoperable services, will enable higher degree of automation and more intelligent applications. For instance, in the case of mathematics, a *semantic* (or *content*) desription, with respect to a merely *presentational* one, opens the possibility to offer major and innovative services like

- Preservation of the real informative content in a highly structured and machine understandable format, suitable for transformation, automatic elaboration and processing.
- Cut and paste of computationally meaningfull expressions (take a formula and paste it into a computer algebra system).
- Automatic proof checking of published proofs.
- Semantical search for mathematical concepts (rather than keywords).
- Indexing and Classification: develop a rich and usefull metadata level specifically oriented to mathematics, and its technological layer.

By joining a content-markup language to the most traditional presentational-makup language, MathML has already undergone a major step in this direction. Still, however, content description is restricted to the case of (common) mathematical *expressions*. On the contrary, in order to reach the ambitious goals described above, we need a clear contenutistic description of *all* aspects of mathematical developments, from the low, "microscopic" level of *proofs*, to the level of meta-data, passing through traditional mathematical structures like definitions, lemmas, theorems, sections, theories and so on.

This is not a mere utopia. For instance, current tools for the automation of formal reasoning and and the mechanisation of mathematics (proof assistants and logical frameworks [7, 8]) already offer *huge* repositories of fully structured, content oriented mathematical information. Unfortunataly, this information is not easily exploitable for the creation of large repositories of structured mathematical knowledge accessible via Web. In fact, libraries in

[1] http://www.w3.org/TR/REC-MathML/

logical frameworks are usually saved in two formats: a textual one, in the specific tactical language of the proof assistant, and a compiled (proof checked) one in some internal, concrete representation language. Both representations are clearly unsatisfactory, since they are too oriented to the specific application: the information is not directly available, if not by means of the functionalities offered by the system itself. This is in clear contrast with the main content-based guidelines of the modern Information Society, and the Semantic Web. The first, mandatory step is thus the direct encoding of the libraries in XML[2]. In fact, a generalized use of the eXtensible Markup Language and its related techonologies provides a solid ground for the ambitious goal of creating and maintaining a virtual, distributed, hypertextual library of formal mathematical knowledge. XML provides a central technology for storing, retrieving and processing mathematical documents, comprising sophisticated Web-publishing mechanisms (stylesheets) covering notational and stylistic issues. By the application of XML technology to the large repositories of structured, content oriented information offered by Logical Frameworks we may really meet the ultimate goal of the Semantic Web, that is to allow machines the sharing and exploitation of knowledge in the Web way, i.e. without central authority, with few basic rules, in a scalable, adaptable, extensible manner.

In particular, the use of XML will provide major benefits in the following fields:

Standardisation Having a common, application independent, meta-language for mathematical proofs, similar software tools could be applied to different logical dialects, regardless of their concrete nature. This would be especially relevant for all those operations like searching, retrieving, displaying or authoring (just to mention a few of them) that are largely independent from the specific logical system.

Publishing XML offers sophisticated Web-publishing technologies (Stylesheets, MathML, ...) which can be profitably used to solve, in a *standard* way, the annoying notational problems that traditionally afflict formal mathematics.

Searching & Retrieving The World Wide Web is currently doing a big effort in the Metadata and Semantic Web area. Languages as the Resource Description Framework or XML-Query are likely to produce innovative technological solutions in this field.

Interoperability If having a common representation layer is not the ultimate solution to all interoperability problems between different applications, it is however a first and essential step in this direction.

Modularity The "xml-ization" process should naturally lead to a substantial simplification and re-organization of the current, "monolithic" architecture of logical frameworks. All the many different and often loosely connected functionalities of these complex programs (proof checking, proof editing, proof displaying, search and consulting, program extraction, and so on) could be clearly split in more or less autonomous tasks, possibly (and hopefully!) developed by different teams, in totally different languages. This is the new *content-based* [1] architecture of future systems.

2 The HELM project

The Hypertextual Electronic Library of Mathematics (HELM)[3] (see [3]) is aimed to integrate the current tools for the automation of formal reasoning and the mechanization of

[2]http://www.w3.org/TR/REC-xml
[3]http://www.cs.unibo.it/helm.

mathematics (proof assistants and logical frameworks) with the most recent technologies for the development of Web applications and electronic publishing. The final aim is the development of a suitable technology for the creation and maintenance of a virtual, distributed, hypertextual library of formal mathematical knowledge, providing a modular environment where information exchange enforces better forms of collaboration. The accomplishment of this project involves the following aspects: *identifying* the mathematical information independent from the specific underlying logical framework; *exporting* to a standardized, open format the knowledge internally encoded in the existent logical systems; developing a suitable set of *modular tools* for analysis, type checking, and interchange of mathematical objects among different logical environments; developing a suitable set of modular tools for storing, cataloguing and querying mathematical documents, via *metadata information*; *rendering* mathematical documents by means of a user friendly interface, taking into account issues such as notational conventions, interactive editing, and easiness of proof understanding.

The present state

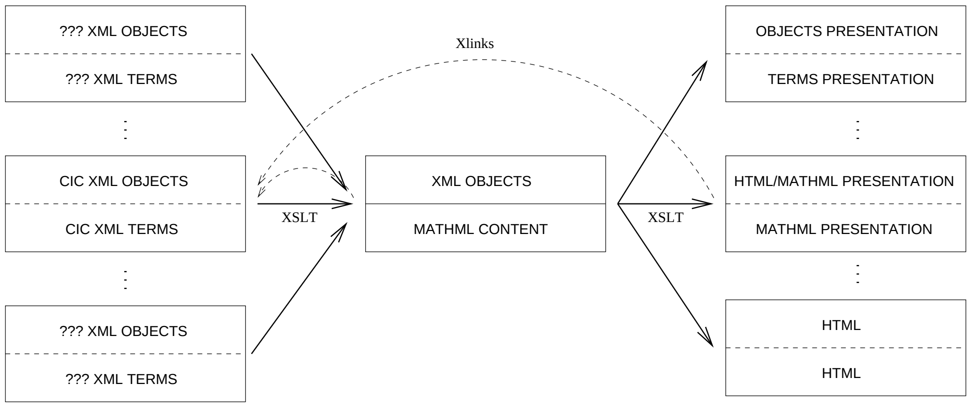
The functionalities of the HELM project already implemented include:

A module for exporting theories from the Coq Proof Assistant [5], a wide-spread and well-maintained application developed at INRIA-Rocquencourt, based on a very rich logical framework called the Calculus of (Co)Inductive Constructions (CIC). We developed a module for exporting the internal representation of CIC terms to XML documents (our module will become a component of COQ-standard distribution starting from Version 7). The module hides to the user all internal representation details of COQ datatypes and structures. The output is a suitable XML-encoding of proof objects expressed in the Calculus of Inductive Construction. In order to be able to process this information you have just to be aware of the formal definition of the Calculus, and not of the COQ-specific internal encoding. The whole standard library provided with the Coq System has been processed in this way, yielding about 128 Mb of XML (25 Mb after compression).

A stand-alone type-checker for CIC objects, similar to the Coq one, but fairly simpler and smaller thanks to its independence from the interactive proof environment. An important point is that objects (definitions, theorems and so on) are type-checked not in the environment in which they were accidentally defined, but in the one made of the only objects on which they recursively depend. This environment is built (and type-checked) "on-demand".

A model of distribution for the library that enables any user with a Web space to publish a document. We have implemented a retrieval system to locate and get an occurrence of a document from a user-supplied set of Web servers.

A set of transformations to render the XML CIC objects to different output formats.



The transformations are specified in the eXtensible Stylesheet Language Transformations (XSLT)[4], that is an instance language of XML for transforming XML documents. For modularity reasons, the transformation phase is splitted into two parts, passing from an intermediate contenutistic language (based on, and suitably extending MathML-content [4]). Stylesheets allow a nice, and user-configurable rendering of mathematical *notation*; the most complex part is proof-rendering, still under development.

The following output standards are currently supported: MathML Content Markup, MathML Presentation Markup (and postscript via the Gtk Widget), HTML.

A Gtk Widget to render MathML Documents (GtkMathView[5]), fully compliant with MathML 2.0 specification. The widget also support translation from MathML to postscript.

A simple XSLT Stylesheet processor (UWOBO[6]) running as a HTTP server. It can be contacted with a proper URL giving the URI of a XML document, a list of stylesheets to be applied, and a list of user-defined parameters. The processor applies the stylesheets in the given sequence (passing the proper parameters) and returns the processed document as the result of the invocation.

The interface to the library. The library could be consulted either with a common Web-browser or with a special-pourpose browser we provide. Even if the Web-browser is not MathML and XLink fully compliant, the user can still visualize the HTML approximation. Our browser, exploiting the peculiar characteristics of the Gtk-MathView widget, is finalized to increase the interaction possibilities, enabled by the contenutistic description of the mathematical informations.

The management and application of the stylesheets is done by UWOBO; due to the light-weight nature of UWOBO, it can be run either on the client machine, or on a dedicated server. Another stand-alone process, providing an HTTP interface, resolves each document logical name (URI) to its physical name (URL) and returns its content, implementing our model of distribution. This process, too, is light enough to run on the client-machine.

The near future

We are going to develop:

A standard language, instance of XML, to describe the structure of a mathematical document, as MathML does for mathematical expressions. This will be possible because the structure is independent from the level of objects of the different logical frameworks. An interesting starting point in this direction is OMDoc (see [9, 10]).

Tools for indexing and retrieval of mathematical documents, based on metadata specified in the Resource Description Framework (RDF)[7]. RDF uses XML to define a foundation for processing metadata, complements XML and provides interoperability between applications that exchange machine-understandable information on the Web.

Tools for the (re)annotation of mathematical objects and terms: the intuitive meaning of these entities is usually lost in their description in a logical framework. Even their automatically extracted presentations in a natural language are often unsatisfactory, being quite different from the typical presentation in a book. We believe that a feasible solution is giving the user the possibility of enriching terms with annotations given in an informal, still structured language. We must remark that for the present, a support to this kind of interactions with a MathML document is lacking.

[4]http://www.w3.org/TR/xslt.
[5]http://www.cs.unibo.it/helm/mml-widget/index.html
[6]http://www.cs.unibo.it/helm/uwobo/index.html
[7]http://www.w3.org/TR/REC-rdf-syntax.

3 Further developments

Our ultimate goal is the extension to other logical frameworks and systems. This will be also an important test bench for the whole architecture.
Another fundamental improvement would be the development of new modular proof engines, supporting step-by-step annotations, so that the proof annotation could respect more closely the human reasoning.

This could be a first move towards the integration of theories developed in different logical frameworks, leading to content centric logical environments.

References

[1] A.Asperti, L.Padovani, C.Sacerdoti Coen, I.Schena. *Content-centric Logical Environments.* Short Communication at the Fifteenth Annual IEEE Symposium on Logic in Computer Science (LICS'2000), June 26 - 29, 2000, Santa Barbara, California.

[2] Asperti, A., Padovani, L., Sacerdoti Coen, C., Schena, I., "XML, Stylesheets and the re-mathematization of Formal Content", Department of Computer Science, Bologna, Italy, May 2000.

[3] A.Asperti, L.Padovani, C.Sacerdoti Coen, I.Schena. *Towards a library of formal mathematics.* Panel session of the 13th International Conference on Theorem Proving in Higher Order Logics (TPHOLS'2000), Portland, Oregon, USA.

[4] A.Asperti, L.Padovani, C.Sacerdoti Coen, I.Schena. *Formal Mathematics in MathML.* Extended abstract at the MathML Internation Conference 2000, October 20-21 2000, Urbana-Champaign, USA.

[5] B. Barras et al., "The Coq Proof Assistant Reference Manual, version 6.3.1", http://pauillac.inria.fr/coq/

[6] Tim Berner's Lee. *The Semantic Web.* W3C Architecture Note, 1998. `http://www.w3.org/DesignIssues/Semantic.html`

[7] G. Huet, G. Plotkin (eds). *Logical Frameworks.* Cambridge University Press. 1991.

[8] G. Huet, G. Plotkin (eds). *Logical Environments.* Cambridge University Press. 1993.

[9] OMDoc: A Standard for Open Mathematical Documents. `http://www.mathweb.org/omdoc/`

[10] The OpenMath Society. `http://www.nag.co.uk/projects/openmath/omsoc/`