Content-centric Logical Environments

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

There is a compelling need of integration between the current tools for automation of formal reasoning and mechanization of mathematics (proof assistants and logical frameworks) and the most recent technologies for the development of web applications and electronic publishing. Currently, libraries in 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 obviously 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 guidelines of the modern Information Society, and its new emphasis on *content*.

The eXtensible Markup Language (XML)[1] is aimed to encode information according to its structure and content. XML, which is rapidly imposing as the main tool for representation, manipulation, linking and exchange of structured information in the networked age, is going to play a pivotal role in the development of a suitable technology for the creation and maintenance of large repositories of structured mathematical knowledge.

The feasibility of describing mathematical structures using a markup language is already testified by the MathML project[2]. The Mathematical Markup Language is an instance of XML for describing mathematical expressions capturing both its notation and content. The emphasis of MathML is just on expressions, while we are also concerned with mathematical objects (definitions, theorems, proofs, and so on) and mathematical documents as well. Therefore, MathML is an essential starting point for encoding terms, which are the real subjects of the mathematical reasoning, while for the objects and the structure of mathematical documents we'll need to develop a new suitable language.

Let us finally remark that the broad goal of the project goes far beyond the trivial suggestion to adopt XML as a neutral specification language for the "compiled" versions of the libraries, or even the observation that in this way we could take advantage of a lot of functionalities on XML-documents already offered by standard commercial tools. First of all, 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. Moreover, if having a common representation layer is not the ultimate solution to all inter-operability problems between different applications, it is however a first and essential step in this direction. Finally, this "standardization" process naturally leads 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-centric" architectural design of future systems.

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

2 The HELM Project

The Hypertextual Electronic Library of Mathematics (HELM)[3] is aimed to provide and support a distributed digital library that could make available in a standard way the already codified mathematical knowledge, and simplify the development of further results in 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[4], a logical system based on the Calculus of (Co)Inductive Constructions (CIC). This module exports the internal representation of CIC terms to XML documents. The whole library provided with the Coq System has been processed in this way, yielding about 64 Mb of XML (2 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 proof engine.

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 following standards are supported: MathML Content Markup, MathML Presentation Markup, HTML. The transformations are specified in the eXtensible Stylesheet Language Transformations (XSLT)[5], that is an instance language of XML for transforming XML documents.

Two interfaces to the library: the first one requires a common web-browser and Cocoon, a XML server-based web-publishing framework. The other one, still under development, needs only a common HTTP server and provides on the client side the XSLT transformation engine and a specialized widget for rendering MathML documents.

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.

Tools for indexing and retrieval of mathematical documents, based on metadata specified in the Resource Description Framework (RDF)[6]. 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.

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.

[3] http://www.cs.unibo.it/~asperti/HELM/home.html.

[4] http://coq.inria.fr/assis-eng.html.

[5] http://www.w3.org/TR/xslt.

[6] http://www.w3.org/TR/REC-rdf-syntax.