

INFORMATION SOCIETY TECHNOLOGIES
(IST)
PROGRAMME

Project IST-2001-33562 MoWGLI

Deliverable n. D4.b
First MoWGLI Prototype
(browsing, rendering, and consultation)

Project Acronym: MoWGLI

Project full title: Mathematics On the Web: Get it by Logic and Interfaces

Proposal/Contract no.: IST-2001-33562 MoWGLI

Contents

1 Overview	3
2 Introduction	3
3 Browsing	3
4 Augmenting the content of the library	4
4.1 Submission of a Coq contribution	4
4.2 Submission of a T _E X article	5
5 Library management	5

1 Overview

The document provides a quick overview of the Web interface to the MoWGLI prototype. Detailed descriptions of the modules composing the prototype can be found in the corresponding deliverables, which are cited in this document when appropriate.

2 Introduction

The MoWGLI project aims at exploiting the possibilities offered by a content-oriented, machine understandable encoding of mathematical knowledge in the creation of a digital library of mathematics. Currently, in accordance with the timetable of the workplan, only primitive functionalities are implemented; the main efforts have been so far devoted to the development of tools for the creation of the content-oriented repository and the implementation of mechanisms for browsing the library.

The library comprises documents from the Coq standard library as well as from several other Coq contributions, and \TeX documents that have been processed with the \LaTeX -to-MathML converter tool (deliverable D4.d). These two kinds of documents (formal mathematical proofs and \TeX articles) have been chosen because they represent two opposite categories: highly structured, automatically generated pieces of mathematical knowledge and handwritten scientific articles in a widespread typesetting markup language.

The first MoWGLI prototype is accessible at the following URL:

<http://mowgli.cs.unibo.it/library/>

Starting from this page, access is provided to subsections that enable the user to perform the following operations:

1. Browsing the content of the library
2. Augmenting the content of the library
3. Managing the content of the library

In the following sections we briefly describe the functionalities offered by the prototype. Detailed instructions can be found directly on the Web site.

3 Browsing

The main functionality that is currently implemented is browsing. Fig. 1 shows a screenshot of the Web interface of MoWGLI in which mathematical formulae are rendered in HTML. MathML output is also supported.

In MoWGLI, there are two main ways for browsing the library, namely via Theories or directly accessing individual Objects, such as Definitions, Theorems, etc.. This is reflected in the organisation of the interface (the top window in Fig. 1): the hierarchy of Theories appears on the left, while the hierarchy of objects is on the right. A Theory in MoWGLI is an arbitrary (structured) collection of mathematical Objects, suitably assembled by some author for presentational purposes and intermixed with explanatory text or figures. Inside a Theory we only visualise statements without proofs: a link to the corresponding proof objects allows the user to inspect proofs, if desired.

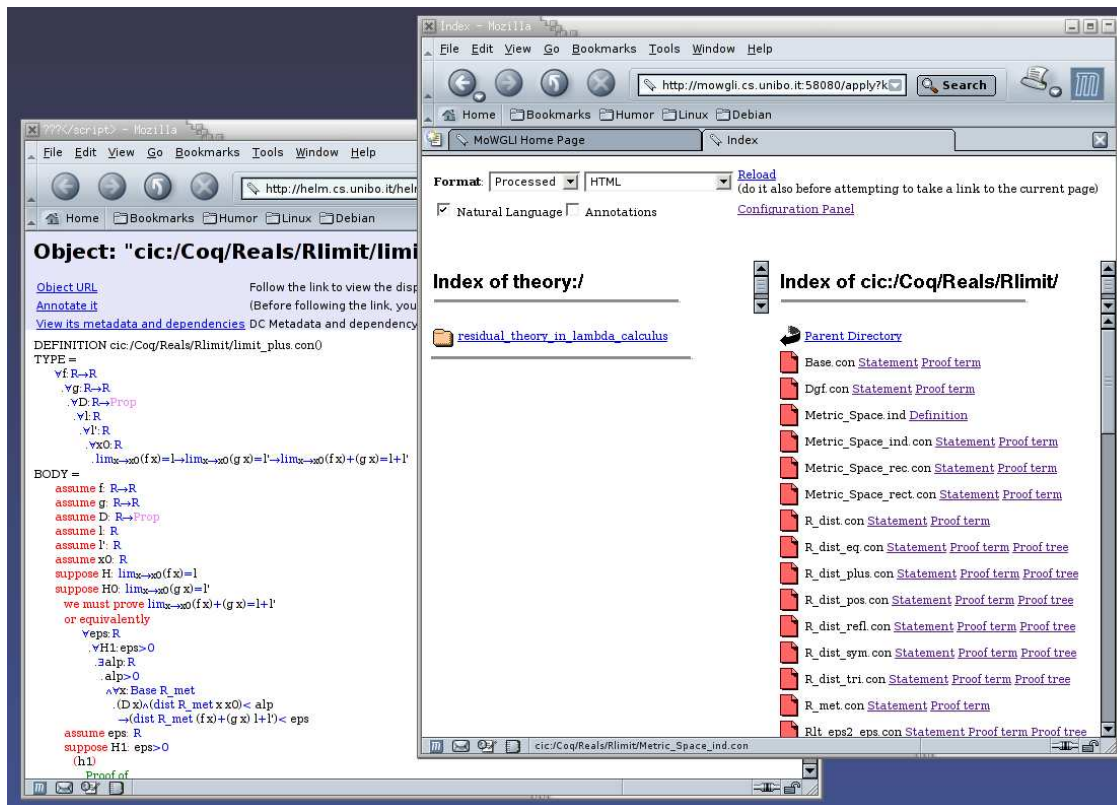


Figure 1: Web interface for the MoWGLI library

The HTML presentational markup interpreted by the browser is generated *on the fly* starting from the logical encoding in XML. This requires a complex transformation of the document; as detailed in deliverables, all the transformations are performed by means of XSL stylesheets (deliverables D2.c-f).

We offer the choice between several different modes for accessing the information, ranging from the raw low-level semantic encoding, to content and presentation formats, such as MathML and HTML.

4 Augmenting the content of the library

The user can either contribute with a Coq development or with a $\text{T}_{\text{E}}\text{X}$ article. The two approaches are described in the next two sections.

4.1 Submission of a Coq contribution

The contribution can be stored in some place that is visible on the Web and that the user can access, or it can be uploaded on the MoWGLI server.

In the first case the user is responsible for exporting the information using a MoWGLI-enabled version of Coq and the only operation left to do is to notify the MoWGLI getter that a new contribution is available and where to retrieve it. This is done by specifying the URL

corresponding to the root directory where the contribution has been exported.

In the second case the user uploads a tar-file containing the `.v` files in the contribution along with a `Make` file. At that point a script on the MoWGLI server performs the following operations:

- creation of a Coq makefile from the `Make` file provided by the author;
- exportation of the objects and of the proof trees from the `.v` files (deliverable D2.a) and storage into XML format into a partition of the server dedicated to users' uploads;
- extraction of metadata from the objects (deliverable D2.g) and update of the database with the extracted metadata;
- notification of the getter. This way the getter gets aware of the new contribution, which becomes available immediately for browsing.

Note that currently proof trees in natural language are displayed correctly only for a small fragment of the library (under `cic:/Sophia-Antipolis/Test/` in the directory structure). We need to patch the exportation module of Coq in order to have correct display of proof trees from the whole library.

4.2 Submission of a \TeX article

The \TeX article must be authored using the macros defined in the `definition` file that is included in the Hermes distribution (deliverable D4.d). This will ensure that MathML Content markup can be generated correctly from the `.tex` file. After submission, a script compiles the `.tex` file and the resulting `.dvi` file is fed into the Hermes converter, yielding a Hermes-specific `.xml` file representing the processed article where the math fragments are encoded in MathML content. Finally, a XSLT stylesheet is applied which converts the Hermes-specific `.xml` file into a XHTML+MathML document that can be viewed with a MathML-enabled Web browser. The `.pdf` version of the article is also created.

All these steps are executed in a Web-accessible directory, so for instance a \TeX article named `example.tex` is accessible after processing at the URL <http://mowgli.cs.unibo.it/hermes/upload/example/>.

5 Library management

This section of the interface allows to remove a Coq contribution from the library and to update an existing contribution with a new version.

When an existing contribution is updated, its content is not removed immediately, but it is saved in case the new version does not compile correctly. Only upon successful compilation the old version is replaced with the new one, otherwise it is restored.