INFORMATION SOCIETY TECHNOLOGIES
(IST)
PROGRAMME

Project IST-2001-33562 MoWGLI

Report n. D0.a
Self-Assessment parameters and criteria

Main Author:
A. Asperti

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 Project Roles and Bodies

The Project has a richly articulated management structure that should ensure, among other duties, the self-assessment and the overall monitoring of the state of advancement of the work.

Following the discussion during the Kick-off meeting in Golm, the main roles have been attributed as follows:

Project Manager Prof.Andrea Asperti, University of Bologna.

Dissem. and Exploit. Manager Prof. Bernd Wegner, Technical University of Berlin.

Work-package Leaders

- **Project Management** A. Asperti (Bologna).
- **State of the art and Req. Analysis** H. Geuvers (Nijmegen).
- **Transformation** C. Sacerdoti Coen (Bologna).
- **Metadata** E. Melis (DFKI).
- **Interfaces** L. Pottier (INRIA)
- **Distribution** M. Kohlhase (DFKI)
- **Testing and Validation** H. Geuvers (Nijmegen)
- **Information Dissem. and Exploit.** B. Wegner (AEI)

The main management bodies are the Project Coordination Committee (PCC) and the Project Exploitation Board (PEB).

The **Project Manager** chairs the PCC. The mandate of the PCC is to represent the Project, report to the Commission, monitor overall performance of the project, ensure accomplishment of the technical objectives, administer project resources and monitor project spending. The PCC is composed by the following people: A. Asperti (chair), M. Kohlhase , H. Geuvers, B. F. Schutz, C. Loiseaux, C. Rideau.

The **Exploitation Manager** chairs the PEB. The PEB is responsible for coordinating dissemination and exploitation activities, promote project visibility and acceptance of project results. The PEB is composed by the following people: B. Wegner (chair), A. Asperti, M. Moschner, H. Geuvers, C. Loiseaux, H. Herbelin.

2 Information Flow

This section shortly describes the main flow of Information inside the project, the tools which have been adopted to facilitate such flows, and the way these tools can be profitably used to monitor and to trace the state of advancement of the work.

Information flow within the Project will be ensured by exchange of internal technical papers, notification of relevant new publications technologies or standards, and reports from external meetings.

All technical documentation generated by the project will be exchangeable in electronic format, using non-proprietary formats (LaTeX, PDF and PostScript), as it has been agreed at project start-up. The project Manager will enforce adherence to these guidelines. Only strictly formal correspondence will be exchanged by ordinary mail and telefax. Urgent correspondence over e-email will be sent with a request for explicit acknowledgement.

2.1 Mailing Lists

Two mailing lists have been created:

mowgli-dev (mowgli-dev@cs.unibo.it). The mowgli-dev mailing list is essentially devoted to technical discussion and development. It is currently composed by 20 members, comprising all developers of the MoWGLI system.

mowgli-adm (mowgli-adm@cs.unibo.it). This is a restricted mailing list, devoted to administration.

Both mailing lists and their respective archives are accessible from the MoWGLI Web-site (in a password protected member area).

2.2 MoWGLI Web-site

The vitality of the Project and all its undergoing activities should be immediately grasped from its Web-site. For this reason, quite an effort has been devoted in the architectural design and implementation of the MoWGLI Web-site (`http://www.mowgli.cs.unibo.it`). Usability and extensibility has been favoured w.r.t. aesthetic principles.

All the gathered information is stored in valid XML files and the Web pages are automatically generated from the XML data by means of XSLT. As a consequence, the DTDs force the consistency and completeness of the entered data and the stylesheets allow to data-mine the stored information in the most effective way, adding new views not conceived when the Web site was designed for the first time. For example, from the list of known events of interest to MoWGLI is easy to automatically extract the list of deadlines, ordered by date.

The site has two main parts, one public and one private. In the public part we find detailed information on the project workplan, all the already released deliverables and publications and up to date information on all the workshops and conferences of interest to MoWGLI users and developers. In the private part (member area) we have on-line archives of the mailing lists, minutes or reports from the MoWGLI meetings and all the other informations that have no public interest.

Thus, for the way it has been conceived, the MoWGLI site itself is a major tool for monitoring and assessing the overall activity of the Project.

2.3 The CVS repository

The Web site is mainly used as a storage of stable important informations. Instead, for the development of the deliverables, our main tool for exchange of information between the participants is the CVS repository, which also have a public read-only area and a private area. In this way not only the final outcome of our work, but also the intermediate steps and important discussions can be recorded and monitored.

Of course the CVS repository is also an essential tool for the collaborative development of all the software components of MoWGLI.

3 Workplan

MoWGLI has a detailed workplan, comprising 29 deliverables (12 software prototypes), and major milestones.

The workplan will be mainly evaluated according to the following criteria:

1. state of advancement of the work w.r.t. the contractual workplan. This evaluation will be mostly based on the respect of the deadlines for deliverables, and their technical quality. Each deliverable belongs to a Work Package and moreover has a specific responsible, which was already identified in the Technical Annex. The responsible for each deliverable acts in conjunction with the WP Leader to organize the work among the partners, collecting contributions, and coordinating the work. All internal communications will make use of the `mowgli-dev` mailing list, so to enhance visibility of the state of advancement of the work.

2. Software compliance to expected objectives and requirement analysis. Each prototype should come along with a suitable documentation providing sufficient ground to allow its technical assessment.

3. Project Retargeting and major technical challenges. MoWGLI is a complex research project, based moreover on a rapidly evolving technology. Most of its prospected architectural and technological solutions are likely to be revised during the project. We also plan to evaluate the Project according to its capacity to (a) cope with the undergoing technological evolution, (b) identify the new major technical challenges, (c) positively and promptly answer to these problems.

More specific and measurable criteria for the single WorkPackages will be listed in Section 7.

4 Project Funding

One of the main activity of the Project Manager is to monitor the project spending, and to attest (as far as possible) the actual costs declared by the partners. To this aim, each site will present a short relation at each periodic meeting (every six months), explaining the state of funding, the costs already incurred with respect to the prospected ones, as well as an overview of the manpower devoted to the performed tasks and the corresponding activities.

5 Project Synergy

One of the most important aspects of self-assessment is, in our view, the degree of satisfaction of each partner in the Project, with respect to its expectations. Although this aspect is hardly measurable, some clear indicators are provided by:

1. cooperation added value. Each partner will provide a short relation aimed to clarify the main reasons for pursuing the collaboration and the added value expected or obtained from it, in terms of knowledge and technological cooperation.

2. tools and knowhow integration. A more precise measure of the previous point can be given by a detailed list of all tools whose development required a tight interaction and collaborations between different partners.

6 Dissemination of Results

A natural way to evaluate the technical quality of MoWGLI is by the list and the nature of the scientific publications related or endorsed by the Project.

This is also one of the main metrics to measure the actual Dissemination of Project results. Dissemination can be further assessed by

1. participation and/or organization of major scientific events;

2. relations with other Projects and/or International Organizations;

3. formal acknowledgement of MoWGLI activity and results by European or international mathematical organizations or institutions.

7 WorkPackages' evaluation parameters and assessment criteria

7.1 Transformation

This work package is devoted to the complex issue of transforming a low level, content description of mathematics (understandable by automatic applications for the mechanization of mathematics) into a human-readable presentational format. It covers both statements and proofs. As general criteria of success we mention extensibility of the transformation process, customization of the output, preservation of the correspondence between content and presentation. We foster the development of transformations that can be exploited in different contexts by different user communities, aiming at a widespread adoption of our methodology and some standardization of the presentational approach.

The transformation will be decomposed in a sequence of intermediate steps, for modularity reasons. All transformations will be implemented by means of XSLT-stylesheets. Stylesheets will be simple, modular, and easily combinable. All the transformation process should be independent from any specific application.

The deliverables fall in three categories:

1. XML exportation module for the COQ Proof Assistant (D2.a).

2. XSL Stylesheets and DTS's (D2.b-f).

3. Tools for automatic extraction of metadata (D2.g).

7.1.1 XML exportation module for the COQ Proof Assistant

The module should support the translation of the standard library of the COQ Proof assistant into a suitable, application independent XML dialect. This also requires the definition of a low-level DTD for the terms of the Calculus of Inductive Constructions (the logical system used by COQ), as well as others DTDs for non-logical information we want to export. At least we need another application dependent DTD for the encoding of proof trees, which are a structured representation of proof scripts.

The task of developing the module will be considered *successful* if we succeed in exporting, in a relatively simple way, all the relevant information saved in the internal encoding of the library. The module will be *very successful* if it will become a standard component of COQ future releases. Finally, it will be evaluated as *extremely successful* it it could induce COQ developers to write a companion *import module*, and to use the XML encoding as an alternative way for storing and exchanging contributions to the library.

7.1.2 XSL Stylesheets and DTS's

The task is devoted to the implementation of stylesheets transforming the low-level logical description of COQ expressions, along with the additional information stored in the proof tree, into a "standard" intermediate, content-level representation such as MathML content, and finally to the expected rendering format.

The task will be considered *successful* if the application of the stylesheets "on the fly" will provide a human readable version of the formal proofs in acceptable times ("acceptable" for a Web interfaces, e.g., say, less than 10 seconds for a complex proof).

The work will be evaluated as *very successful* if we may induce COQ users to use MoWGLI's stylesheets as the most natural way for publishing their contributions on the web.

Finally the stylesheets will be *extremely successful* if other groups will start using them and/or contributing to their developments.

7.1.3 Tools for automatic extraction of metadata

Most part of MoWGLI's indexing and retrieving functionalities will be based on the definition of a sophisticated metadata model and its exploitation via a suitable query language. The main bulk of these metadata will be automatically extracted from documents, taking advantage from the content encoding of the mathematical information. These metadata are meant to give an approximation of the actual content (such as, say, list of identifiers in critical positions inside statements or expressions) suitable for fast searching and retrieving operations.

Extraction of these metadata is considered as a batch process to be periodically applied to the whole repository of mathematical documents. Given the dimension of the library, the first evaluation parameter will be the possibility to compute the whole set of metadata in a reasonable amount of time (e.g. within a night). A more detailed evaluation may be only given in conjunction with other metadata tools and the query system.

7.2 Metadata

The precise definition of metadata and their actual Markup Model are essential aspects for implementing the main functionalities of the library, and especially for archiving, searching and retrieving issues. The work will be articulated in two main, almost sequential, tasks:

T3.1 Use, meaning and classification. Delineation of the basic intelligence to be considered for encapsulation in metadata, in order to meet the needs delineated during Requirement Analysis.

T3.2 Modeling. This is devoted to the definition of a precise markup model. To this aim, we plan to use the Resource Description Framework of W3C.

We shall jointly develop a unified way of allowing users to query the metadata model. Our solution should be general and flexible enough to return mathematical equations (in a format chosen by the user), proofs, definitions, whatever. It should basically be an API (application programming interface) that could be implemented in a variety of ways. A web publisher could, for example, provide a graphical search interface that hides the complexity of the query language behind user-friendly buttons and options, in the end composing a query to the search engine. Other Web sites could do their own implementations that may look very different. Similarly, the way the web site does the search in response to the query would be very document dependent. The query language itself can be something the whole community agree on and share.

The metadata system will be considered *successful* if, on typical queries based on content patterns, it will allow retrieving of a sensible set of documents in reasonable time (few seconds).

It will be evaluated as *very successful* if typical users of the library (e.g. COQ users, or, in a different field, contributors of AEI's *Living Reviews in Relativity* Journal) will start to extensively use it as their favourite mechanisms for retrieving information in the mathematical repository.

Finally, it will be *extremely successful* if we get some international recognition of our work, either by wide-spread adoption of the system, or by "sponsorization" (in the form of notification and advertisement) of major international mathematical organizations (such as AMS or EMS), eventually leading to a standard.

7.3 Interfaces

This Work-Package is devoted to the design and the implementation of the interfaces to the library, covering rendering, browsing, searching and retrieving functionalities. Some additional functionalities for authoring purposes (such as computer assisted annotation of proofs) will be also taken into account.

The main deliverable consists of the Web-based browsing and consultation interface. Additionally, we will provide the following prototypes addressing tasks that cannot be easily implemented using common Web browsing applications:

1. MathML rendering engine;
2. Ad-hoc user interface to the COQ system and library;
3. LaTeX-based authoring tool.

Other prototypes might include a tool for the manual editing of metadata, and a user-friendly interface for the query language.

7.3.1 Web-based browsing and consultation interface

One of the key points of the MoWGLI project is to provide access to repositories of mathematical information by means of standard, commonly used Web browsers. For those tasks that cannot be effectively addressed using standard applications only, such as sophisticated rendering, interaction, and editing operations, we will develop dedicated interfaces, discussed in the sections that follow. The Web interface, as well as the dedicated applications, will be conceived as the integration of components already developed for the other MoWGLI tasks, augmented with the requested features. Among the main components that must be exposed in the Web prototype there are the transformation tools, which render the proofs, and the query engines.

In addition to the validation criteria of its components, the Web interface must stand out for its user friendliness and for the smooth integration of its parts. In particular, the interface will be considered *successful* if it provides uniform access to the different parts of the MoWGLI library, ranging from formal mathematical developments to the physical papers. The interface will be *very successful* if the interface represents a real enhancement with respect to current tools for the publication and exploitation of mathematical documents. The interface will be *extremely successful* if the MoWGLI interface, and its related tools, will be adopted by a large group of users, possibly outside the communities the project originated from.

7.3.2 MathML **rendering engine**

Our privileged rendering language will be MathML, which is likely to be rapidly adopted as the main language for representing mathematical notation on the Web. Unfortunately, no adequate support for this language is currently available, and we have planned to develop our own rendering engine inside MoWGLI. The tool is conceived as a software component whose main core is a platform independent C++ rendering engine for MathML presentation markup. Different interfaces can be provided depending on the platform of interest. We plan to provide at least two interfaces, for GTK and PostScript.

The GTK interface will be used to render MathML inside GTK applications. It will allow users to view MathML, interact with the markup (selection, activation of ¡maction¿ elements) and possibly support editing. Given the wide range of potential applications, interaction semantics should be easily customizable by the programmer. The GTK interface could be used to provide MathML support in any math-oriented application (calculators, converters, proof-assistants).

The PostScript/PDF interface will allow quality rendering of MathML for embedding in larger documents and printing.

We will consider the tool *successful* if it is compliant to the MathML specification, which can be verified with respect to the standard MathML test-suite defined by the MathML Working Group. The tool will be considered *very successful* if it supports a number of different rendering interfaces and platforms, and it can be easily embeddedable within larger applications. Finally, the tool will be evaluated as *extremely successful* if adopted in other math-oriented applications developed by external teams.

7.3.3 Ad-hoc user interface to the COQ system and library

We intend to provide a working environment able to handle the documents that will be produced from COQ formal developments, whether they would be available from the Internet or only available on the user's machine. This tool should display data in a style that is as close as possible to what would be available to other Web users equipped with off-the-shelf web browsers (possibly complemented with the MathML rendering engine described in the previous section). Connections to the tools for search and retrieving should be provided using regular-style dialog elements. This working environment will also provide editing facilities and connections to the COQ tool, so that it should exhibit authoring capabilities.

The first criterion for evaluating the tool will be its capability to view the data attached to all parts of the MoWGLI digital library related to COQ formal proofs. The second criterion of success will be that this tool can be used to produce Mowgli documents from COQ formalized proof. A third criterion would be that this interface also integrate capabilities for Mowgli documents that are not related to COQ formal proofs.

7.3.4 LaTeX-based authoring tool

This tool should provide functionalities for creating content-based mathematical information from standard digital repositories by means of a suitable LaTeX-based authoring system. The system should also support manual insertion of metadata. A first prototype will be available since September 2003, and will be first tested among the authors of the electronic review journal, *Living Reviews in Relativity*, edited by Albert Einstein Institute (AEI) in Golm (Germany), which provides refereed, regularly updated review articles on all areas of gravitational physics. The advanced prototype, integrated with other MoWGLI features (especially for indexing and retrieving), should serve as a showcase to demonstrate how content-mark-up in mathematics improves the usability and information depth of electronic science journals.

The success of the tool will be primarily measured by the number and quality of enhanced functionalities it could pave the way to. A second, essential parameter will be its actual use by authors of Journal Articles. In particular, the tool will be regarded as *very successful* if almost the totality of new submissions to *Living Reviews in Relativity* will make use of the authoring system. The tool will be *extremely successful* if adopted by other electronic Journals.

7.4 Distribution

This WorkPackage is devoted to the overall architectural design of the distribution model, its implementation and integration with the consultation engine. The work is articulated in three, conceptually sequential tasks:

T5.1 Architectural Design of the Distribution Model. The big issue is to find the right compromise between two opposite requirements: *distribution* (in the sense of the Web: few rules, no central authority) and *coherence* (coherence between different copies of a same document and global management of the library as a single rational development). Other architectural problems to be solved are the management of Uniform Resource Identifiers, their mapping to Uniform Resource Locators, and the integration of databases in the distribution model. The final aim is to have a *physically* distributed library with a single *logical* view and significant searchability features.

T5.2 Prototype implementation. First prototyping implementation of the distribution layer.

T5.3 Integration with the Consultation Engine. First implementation of the library as a distributed repository. Distribution should be completely transparent to users of the Consultation Engine.

The task will be *successful* if it will support the distribution of MoWGLI's sample library of documents among the different partners, and its distributed management. It will be *very successful* if we could convince other interested parties to publish documents using MoWGLI's technology at sites external to MoWGLI's Consortium.

7.5 Testing and Validation

The WP intends to measure the system suitability and scalability and the satisfaction level of users with the service. Large scale testing and validation will start after the release of the first MoWGLI prototype, at month 18. We shall consider three main validation tests:

T6.1 Education. Full development of a fragment of the library covering a typical undergraduate course in algebra of analysis. The development of this course will be adjusted to match the requirements of chosen course and shall be tested at the same time as a normal course. The usage of the learning environment making the most use of the content-encoding should be considered *satisfying* if the content-oriented features are usable by the students. It shall be considered *successful* if it allows a subset of the students that spent time using the tool to perform better in exams. It shall be considered *very successful* if the tool is adopted by other universities and existing content is re-used there.

T6.2 Certified code. The aim is to be able to present the formalization and the demonstration of some security properties related to the code embedded into a smart card. The tools developed in MoWGLI will be evaluated according to the degree of readibility of the formal development, to be assessed by the client company commissioning the certification work.

T6.3 Electronic Publishing. The aim is to test the LaTeX-based authoring tool and to demonstrate how an electronic physics journal benefits from the exploitation of content markup in journal articles. One of the main activity of this task will be the assesment of the evaluation parameters already detailed in Section 7.3.4.