WIRIS OM Tools: a Semantic Formula Editor.

Daniel Marquès, Ramon Eixarch, Glòria Casanellas, Bruno Martínez

Maths for More S.L. , Rambla de Prat 21 1r 1a 08012 Barcelona, Spain
[info@mathsformore.com]

Tim J. Smith

School of Informatics, University of Edinburgh
[tim.smith@ed.ac.uk]

2000 Mathematics Subject Classification. 20 68-04

Abstract

With the increasing reliance on computers for the automatic processing of information a new method is needed for editing mathematical formulae. We are used to WYSIWYG editors that produce beautiful presentations of formulae and store the typesetting primitives rather than the meaning of the formulas. However, new services such as database searching or calculation web-services work best if they have access to the semantic information behind a formula. This can only be done with a new generation of formula editors. In this paper we present WIRIS OM Tools [17], a semantic oriented formula editor which addresses these concerns. It is based on the OpenMath language and a suitable transformation process between OpenMath and MathML expressions. Additionally, this approach adds new features for the users such as error, type and syntax checking. The editor is currently being used in the LeActiveMath and WebALT projects.

1 The goal

The large majority of currently available formula editors are able to edit and present formulas in a way that is easily readable to users. For automatic computer processing, the resulting formulas turn out to be ambiguous or, simply, there is no interpretation to them since they are based of the typesetting result only. In particular, one would like to have a single formula editor as input interface to a broad spectrum of Computer Algebra Systems [10], [13], search repositories with mathematics [9], and online tools (like MediaWiki's and blog's) for sharing math content over the web. Presentation-based editors do not fulfil the fundamental requirement of such tasks: storing a representation of the formula that can be manipulated and computed.

A solution is to develop formula editors that output the mathematical fragments written in a suitable markup language. Such markup expressions should contain unambiguous semantic information. Content-MathML and OpenMath [8] both serve to this purpose.

We propose a solution based on OpenMath, whilst the conversion to Content-MathML is left to the user and can be carried out using a standard XSLT [1]. WIRIS Editor [16], a mature presentation editor, has been adopted to produce content markup.

[1]http://www.w3.org/Math/Software/mathml%5Fsoftware%5Fcat%5Fstylesheets.html

2 State of the art

The widely used general purpose WYSIWYG formula editors are presentation oriented, for example MathType [7], WebEQ [15], Formulator[1], MathCast[6] and TechExplorer [12]. Other editors appear integrated in Computer Algebra Systems, word processors or scientific applications and are not the focus of this article.

Some projects attempt to obtain content formulas using a parser or XSLT transformations to convert presentation to content markup. In general this is a difficult task and there is probably no universal solution. Moreover, errors detected in the input formula by the parser cannot be displayed in the original editor.

Content oriented editors are difficult to find. The few we have found are at a developmental stage: JOME [2], and IBM MathML Expression Editor [11], a TechExplorer application. The latter is a template-based editor creating Content-MathML. In a template-based editor, the user interface contains a palette with atomic formulas. Once one of these atomic formulas is selected, it is added to the current formula. Afterwards, only the blanks are editable. Formulas can be built up recursively using new atomic formulas. However, this kind of editing results in a bad experience for the user. Our approach adopts a presentation style of editing to improve the user experience.

3 WIRIS OM Tools

Some of the system requirements to achieve the main goal of developing a content-oriented formula editor based in OpenMath are listed below.

1. Enhance user experience in the creation of formulas with a WYSIWYG interface. Content oriented editors have often not offered WYSIWYG interface but linear input interfaces.

2. Remind the user which constructions are available by a *configurable* palette with atomic formulas.

3. Allow keyboard input for numbers and basic operators, including constructions like fractions, roots, big operators and matrices which can be typed using the *linear input*. It is important to allow keyboard input process in order to speed up the formula inputting process.

4. Support OpenMath extensibility by allowing new symbols to be added to the editor. This is a key feature for a content-oriented editor. In several contexts new OpenMath CDs are created to support a project's requirements. As an example, the WebALT project [14] has created new OpenMath CDs to support mathematics exercises in natural language.

5. Toolbar configurability. The palette must be configurable by the user in order to adapt it to various contexts. The toolbar required for a Calculus course may be different from the expected at a Linear Algebra course.

   The toolbar must be configurable in two ways: first of all, allowing different icon presentation in folders and finally, allowing users to include any OpenMath symbol in the palette with the same quality as the original symbols supported.

6. Perform error checking in real time. This feature is missing in presentation formula editors and is only possible with semantic information. That feature is useful to avoid unnecessary calculations when the editor is integrated with Computer Algebra Systems or e-Learning platforms [3].

7. Support cultural variations in formula presentation. A list of issues that must be supported are

   [a] Allow different symbols for decimal separator;

   [b] Allow different symbols for argument separator;

   [c] Expand especial symbols to define variables.

8. Support web-application and a variety of platforms using the Java plugin technology.

9. Include extra standard editing features like copy, paste, undo and redo. These added features are available not only inside the editor but also within the web applications in which the editor is embedded.

10. Embed true type fonts and good support for font typesetting to get high quality rendering of formulas. In order to achieve high quality results it is a key point to be able to offer high quality fonts such as True Type. Since these components can be embedded in a variety of tools they must be able to adopt the formulas look and feel to the whole application look and feel.

The suite of software components WIRIS OM Tools, originally developed to edit OpenMath in WYSIWYG mode for the LeActiveMath project [3], fulfills all the requirements set for a content-oriented formula editor. More details on the solution offered for all the previous items can be found in LeActiveMath Deliverables documentation [5]

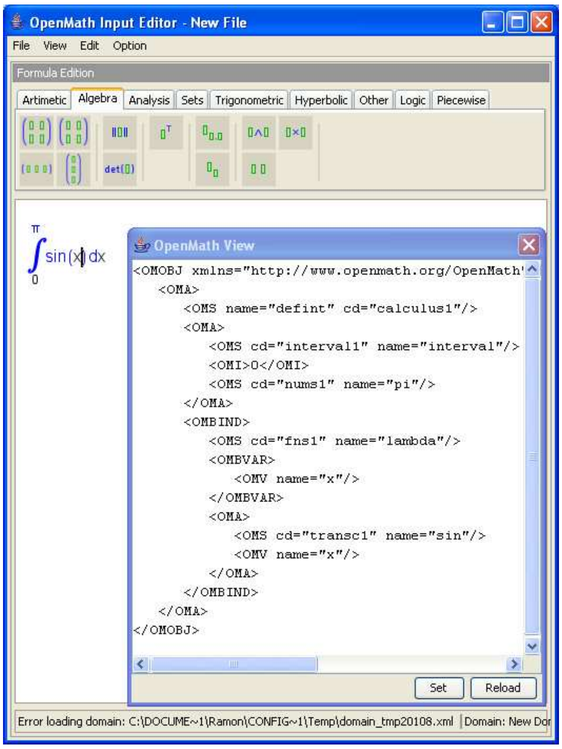

Figure 1: Input editor and OpenMath associated with the formula.

A new concept has been defined in order to manage the translation from MathML to OpenMath, the symbols extensibility and the toolbar configuration. This concept is referred to as the Domain. A new XML language has been set up to determine a Domain.

A Domain is a description of the relation between OpenMath symbols and their MathML presentation. That description allows the system to translate from Presentation MathML to OpenMath. Moreover, the Domain file also describes the toolbar of the editor. Since there is no privilege given to official OpenMath symbols a Domain file can support the description of a toolbar and MathML-OpenMath relation for any OpenMath symbol allowing the system extension.

The main software components are:

Formula Editor Full solution for editing OpenMath in a WYSIWYG mode. It is a Java component that can be embedded in web-based solutions or Java applications to provide formula creation for users.

Formula Component Low level component for editing formulas and script MathML.

Phrasebook OpenMath to/from MathML translator. This is the key component that transforms Presentation MathML to OpenMath based on the information described in the Domain file.

Domain Editor WYSIWYG editor of Domain files that are used to define the palettes. Since the Domain file is the key component of this system we have developed an application that, with a WYSIWYG interface, permits the creation of Domain files.

Font Editor Basic font typesetting editor in True Type format.

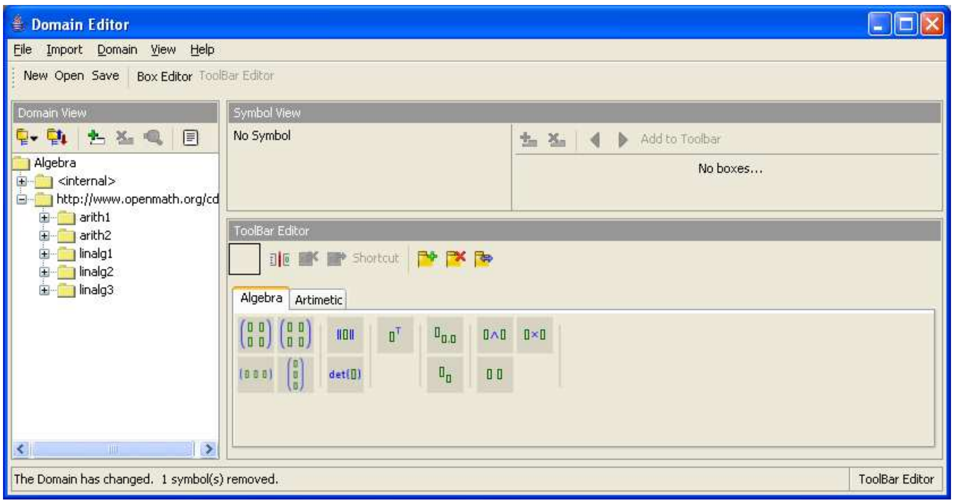

Figure 2: The Domain editor used to produce Domain files.

The main steps in the editor workflow are the following. First, the mathematics edited in the WYSIWYG mode needs to be associated with markup tags. We have chosen to use Presentation MathML as it is the most used standard and that allows interoperability with other tools.

Since we want to have OpenMath expressions we need to translate MathML to and from OpenMath. The translation process is done by the Phrasebook component and defined by the Domain file. We cannot jump over the presentation language precisely because one presentation formula may have several content interpretation in various contexts.

Finally, to ensure a pleasurable user experience we need ease the editing process by organizing icons into toolbars. The definitions for these toolbars are also included in the Domain language.

The translation of the formula representation to OpenMath is enhanced to make error checking in real time possible. This means that, at the same time the user is typing a formula, its correctness is tested in the background and the errors are highlighted. The two kind of errors detected are syntax errors and type errors. Syntax errors include missing arguments for operations, unclosed parenthesis or the use of non existing symbols. Type checking includes testing the arity of functions and arguments with a wrong type. For example, if an symbol is defined to receive for two Numerical Values as arguments there will be an error warning if one of the arguments is a Matrix. The type checking system is based on the STS signature for OpenMath symbols.

4 Usage in LeActiveMath and WebALT project

The WIRIS `Formula Editor` is integrated into LeActiveMath system as an applet. It is used mainly inside the exercise subsystem to facilitate formula editing for learners in exercises.

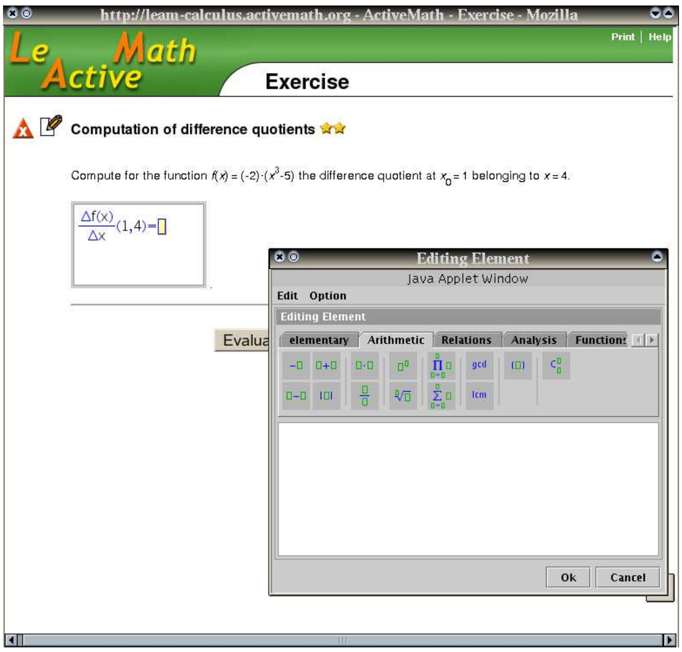

Figure 3: Editing complex formulas with blanks in LeActiveMath exercises

It is also used by the search components of LeActiveMath (see [4]) and the exercise repository in order to allow for semantic search of formulas. This integration happened problem free and is depicted in Figure 4.

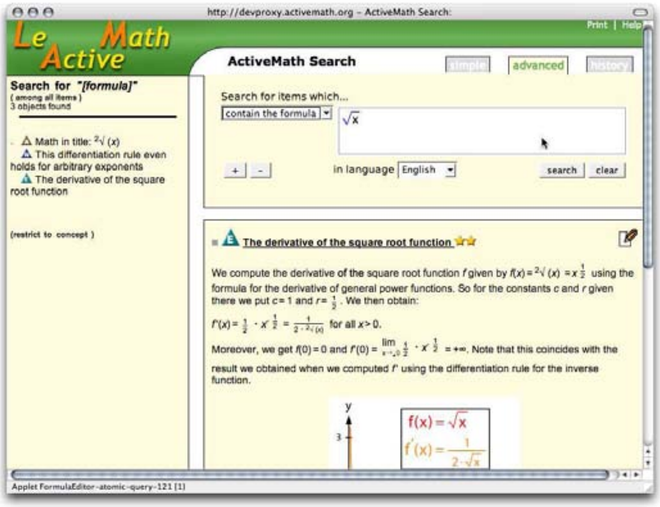

Figure 4: Searching for a formula using the WIRIS `Formula Editor`

Finally, the exercise authoring tool uses the `Formula Editor` to allow easy authoring of formulas in feedback text and CAS queries.

WebALT project aims to develop software tools that allow the creation of mathematical sentences, mainly for exercise questions and answers, in multiple languages automatically.

WebALT TextMathEditor is a piece of software that allows the construction of mathematical sentences that follow a predetermined grammar. It is an OpenMath oriented application currently under development. WIRIS Formula Editor has been integrated in the TextMathEditor as a formula editor. Some of the advanced features of the WIRIS OM Tools such as error checking have been used to enhance TextMathEditor.

5 User-testing

To assess the benefit of the WIRIS OM Tools user tests were performed by the Le Active Math University-level evaluation team (University of Edinburgh, UK). The goal of the Le Active Math project is to develop a web-based eLearning system for the teaching of differential calculus both in formal learning environments, such as school classrooms and university, as well as at home as part of a student's homework or self-motivated learning [3]. To achieve this goal Le Active Math emphasises the role of intelligent feedback, support, and guidance during a user's interaction with the system. One of the key components of this feedback is the ability for the Le Active Math system to understand the semantics of learning objects including mathematical formulae. The WIRIS OM Tools provide Le Active Math with the necessary bridge between OpenMath and MathML and the Formula Editor provides the user with a way of constructing semantically-tagged formulae.

User tests of the Formula Editor were performed to assess the usability of the Formula Editor and user's perception of its benefit and comparison to existing systems. Users performed a series of tasks using Le Active Math including several exercises that required formulae to be either input linearly using the keyboard or using the WIRIS Formula Editor. Their performance was monitored and they were asked to complete a questionnaire about the system after using it.

The Formula Editor exceeded user expectations. Before using the editor only 69% of users could see the benefit of using such an editor and after using WIRIS 73% of users said they would use the WIRIS Formula Editor in the future. Users rated the WIRIS Formula Editor marginally more practical (mean rating of 4 on a 5 point Likert scale with 5 as high) than previous editors they had encountered (mean rating 3.89). When probed, most users mentioned Microsoft Equation Editor as the editor they had most experience of. When asked to rate how easy they found inputting formulae using typed linear input compared with using the Formula Editor users rated the Editor as easier (4.36 out of 5) compared to typed input (4.09). Users also rated the Formula Editor as very useful for inputting formula (4.91 out of 5).

Taken in combination these ratings indicate that users perceived the benefit of the WIRIS Formula Editor and would like to use it in the future. This benefit could also be seen in user's performance when using the formula editor. Without any introduction or instruction most users were able to immediately start constructing complex formulae using the editor. Most users understood how the templates could be combined to create complex formulae and commented that they preferred this method to typed linear input. Users also liked the fact that the Editor allowed them to input formulae in a typed linear fashion and automatically convert it into the graphical format. This allowed users to transfer their existing techniques of inputting formulae direct to the WIRIS editor.

Once users were familiar with the Formula Editor's templates users were much quicker at inputting complex formulae using the Editor than typed linear input. This efficiency was also aided by the intelligent syntax checking and feedback provided by the editor. By highlighting syntax errors during a formula's construction the system guided users towards the syntactically correct constructions. If the user attempted to use an incorrect formula the editor would provide detailed information about the error. Such syntactical checking is essential within Le Active Math as the user's progress is monitored based on their answers to exercises. By ensuring syntactical accuracy before a user can use a formula to answer an exercise, the WIRIS Formula Editor helps Le Active Math distinguish between syntactical and conceptual errors. Only once conceptual errors are identified can LeActiveMath provide targeted tutoring.

6 Conclusions and future work

WIRIS OM Tools provide an editor for content (OpenMath) as usable as the currently used representation editors. In fact, a user will never notice that he is using a content oriented editor.

Considering solely the editing functions, content oriented editors such as WIRIS OM Tools offer the benefit of syntax and type error checking at real time. The flexibility of the editor allows it to cater for any mathematics topic. Such configurability of the editor is essential for the accomodation of different meanings of an operator or expression across different arithmetic practices.

Development of the WIRIS OM Tools is on-going and we plan to resolve some of its current limitations. For instance, the current content editor lacks some desired presentation features. When rendering a formula, symbols are represented depending on the current configuration of the editor. For example, divisions could be rendered as fractions ($\frac{1}{2}$) or use the operator '/' (1/2) but a single formula cannot contain both fractions and the '/' operator. Features like text attributes (colours, font size and style) are also missing. We plan to solve these problems by providing a standard set of new symbols that allow the user to configure the presentation aspects of OpenMath. With the addition of these features we hope that users will be as satisfied with the presentation of their constructed formulae as they are with the rest of the features of the WIRIS OM Tools.

Acknowledgements

We are grateful to the LeActiveMath and WebALT projects for having trusted in our development team and for the support and testing work in the project development.

References

1. Formulator. Home page at `http://www.hermitech.ic.zt.ua/projects/formulator/`
2. Laurent Dirat, Michel Buffa, Jean-Marc Fedou and Peter Sander. JOME: OpenMath on the Web. Available at `http://www.tech.plym.ac.uk/maths/ctmhome/ictmt4/P55_Dira.pdf`
3. LeActiveMath. Home page at `http://www.leactivemath.org`
4. LeActiveMath Partners. D15: Enhanced Dictionary LeActiveMath Deliverable D15, The LeActiveMath Consortium, July 2005.
5. LeActiveMath Partners. D38: Input Editor LeActiveMath Deliverable D38, Ramon Eixarch, Daniel Marquès, Gloria Casanellas, Bruno Martinez, December 2005.
6. MathCast. Home page at `http://mathcast.sourceforge.net`
7. MathType. Home page at `http://www.dessci.com/mathtype/`
8. OpenMath. Home page at `http://www.openmath.org`
9. Paul Libbrecht and Erica Melis. Semantic Search in LeActiveMath. Proceedings of the WebALT 2006 Conference. Available at `http://www.activemath.org/publications/Libbrecht-etal-SemanticSearch-WebALT2006.pdf`
10. Research Institute for Applications of Computer Algebra. Phrasebooks web page `http://www.mathdox.org/products/phrasebooks`
11. Samuel S. Dooley. Bringing MathML Content and Presentation Markup to the Web with the IBM MathML Expression Editor. Available at `http://www.mathmlconference.org/2002/presentations/dooleyxml/`
12. TechExplorer. Home page at `http://www.integretechpub.com/techexplorer/`
13. The MONET Project. Home page at `http://monet.nag.co.uk`
14. WebALT. Home page at `http://www.webalt.net`
15. WebEq. Home page at `http://www.dessci.com/en/products/webeq/`
16. WIRIS. Home page at `http://www.wiris.com`
17. WIRIS OM Editor. Test page at `http://www.wiris.com/demo/omeditor/`