Semantics for Web-Based Mathematical Education Systems

The ActiveMath group:
Erica Melis, Jochen Büdenbender, Georgi Goguadze, Paul Libbrecht, Carsten Ullrich
melis, jochen, george, paul, cullrich @activemath.org
Universität des Saarlandes, D-66123 Saarbrücken, Germany

ABSTRACT

Web-based user-adaptive learning environments suggest a semantic knowledge representation that can be reused in different contexts. Moreover, if these educational systems employ external service systems for support or for exploratory activities, the semantic representation is a basis for the interoperability of the service systems and for machine-understandable data. ActiveMath is such a learning environment for mathematics. We show what its annotated semantic knowledge representation, extended OMDoc, is like and how it is used. We also discuss the current bottleneck of authoring. Since mathematicians are mostly familiar with authoring LaTeX rather than semantic XML, ActiveMath offers a LaTeX2OMDoc tool. As compared with the direct OMDoc authoring which is not yet visually supported this has pros and cons.

1. INTRODUCTION

Many educational systems and on-line documents have been produced in recent years. Since the encoding of the domain knowledge for a learning environment is a very expensive and time-consuming task, *reusability* of the encoded knowledge in different contexts and for different functionalities is desirable. Therefore, the representation needs to incorporate an ontology of the domain or, even better, a unique and extensible semantics of the domain concepts and their various relationships. Similarly, *inter-operability* is a prerequisite for multiple services used in education systems that can access and work with common knowledge sources.

For Semantic Web applications, mathematics is a good field to experiment with because it is largely formalized and has a clear fundamental semantics independent of presentational issues and because mathematics is a relatively well-structured field. For mathematics, an ontology needs to be enhanced by real semantics because mathematical knowledge is inherently different from its presentation (e.g., its printed version). Different presentations can mean the same thing, e.g., $\frac{1}{2}$ or 1/2. Conversely, the same presentation can mean different things in different contexts, e.g., $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ is false in elementary algebra but true in the theory of quadratic residues.

Now, our learning environment ActiveMath [8] is a Semantic Web application for mathematics learning. Its knowledge representation is separated from its functionalities. Its knowledge representation meets the above requirements for mathematical content representations and those for educational applications that include the above mentioned reusability and inter-operability as well as the representation of pedagogical information.

ActiveMath' knowledge representation is based on OpenMath [3], a general, standardized, semantic XML-representation for mathematics. ActiveMath' functionalities require additional information to be encoded into the knowledge representation, e.g., structural information such as is-a-definition and pedagogical information such as the difficulty of an exercise.

This article shows how Semantic Web issues such as machine-understandable representation, reusability, extensibility, and migration of other representations are tackled in ActiveMath. It focuses on the knowledge representation and its current authoring. It summarizes which information is represented in the OMDoc-language which is an extension of OpenMath. It describes and substantiates the extensions we have added for the educational and other purposes of ActiveMath. It discusses how content is authored presently in a situation, where tools for semantic representations are emerging only and where the habits of authors still oppose such an encoding.

2. SEMANTIC REPRESENTATION

Although today's most common representation for knowledge of web-based systems is the syntactic markup language HTML, for a meaningful reuse in different contexts and knowledge sharing the XML representation is essential and the RDF [1] framework with data representing relations between elements can serve as a basis for building an ontology.

2.1 Semantics in Mathematical Knowledge

OMDoc has evolved historically as a standard for mathematical knowledge representation, which we decided to use for our educational system. Because of this history several features have still to be adapted to semantic Web developments, e.g. RDF. However, a big advantage of using OMDoc is its truly semantic flavour.

[1] http://www.w3.org/RDF/

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Semantic Web Workshop 2002 Hawai, USA
Copyright by the authors.

What is the information and elements needed for mathematics? To ensure the inter-operability of mathematical systems, a keyword-annotation that might suffice for simple search functionalities is not sufficient anymore. A machine-readable input for mathematical systems such as Computer Algebra Systems (CASs) and theorem provers requires a mathematical semantics. That is, to provide a basis for multiple systems, the actual mathematical objects/formulas have to be represented. Candidates for the representation of mathematical objects and concepts are the `XML`-languages `OpenMath`[2] and content-`MathML`.[3] `OpenMath` is a (European) standard for the semantic representation of mathematical formula expressions. It semantically defines a set of mathematical symbols in the `OpenMath` content dictionaries and can import content dictionaries into others.

However, an extension of `OpenMath` is needed because (1) an ontology based on `OpenMath` lacks most relations except of theory-inclusion, (2) the `OpenMath` content dictionaries are incomplete and a simple extension mechanism is needed, (3) `OpenMath` has no means to structure the content of a mathematical document by dividing it into its logical units such as "definition", "theorem", and "proof".

For these reasons, `OpenMath` has been extended to `OMDoc` [6] which includes structure markup for mathematical concepts such as `definitions` and `theorems` and for other items such as `examples`, `exercises`, and `elaborative texts`. It also allows to define new symbols. `OMDoc` items may contain metadata, formal elements, textual elements, and references. References can be concept identifiers, URLs, and additional code elements. Metadata in `OMDoc` represent legal information compliant to Dublin Core metadata [11] and `extradata` element for metadata extensions. `OMDoc` allows to represent some mathematical dependencies: morphisms between theories, equivalence of definitions, proof-for. Implicitly it contains a dependency of symbols by the occurrence of symbols in another symbol definition.

For our application the `OMDoc` metadata and relations between elements are insufficient, because, on one hand, ActiveMath needs a pedagogical ontology. For its use in learning environments, we have extended `OMDoc`. On the other hand, relations between elements such as mathematical dependency, corollary of a theorem, similar examples, counterexample for a concept, which we introduce are also general for the field rather than due to the educational application,but are not present in `OMDoc`. Some of our extensions are not specific for tutorial applications, such as technical requirements[4] and bibliographical references.

2.2 Pedagogical Knowledge

The extensions described in the following are motivated by the tutorial application and generally applicable for learning systems rather than specific for a mathematics system.[5] They include, among others

- pedagogical *properties* such as difficulty. They are relevant for ActiveMath' user-adaptivity because these metadata allow to present materials that fit the current learning situation;
- pedagogically motivated *relations* among the different pieces of knowledge such as is-prerequisite. This information can be used, e.g., to present the learner prerequisites for understanding a concept, to generate links, and to generate an adaptively chosen and structured learning content.

On the one hand, metadata standards for learning resources [6] contain too many metadata which are not relevant for our purpose. On the other hand, adaptively presenting content requires some metadata which are not yet specified in LOM. This is not surprising, since IMS[7] will be soon extended by Educational Modelling Language (EML)[8] metadata. Therefore, the ActiveMath metadata extensions of the `OMDoc` DTD include some metadata from LOM as well as some others. For example, we extend the list of possible values of the `type` of attribute of LOM metadata element `relation` as described below. More detailed, the pedagogical metadata defined in the ActiveMath-DTD are `field`, `abstractness`, `difficulty`, `learning-context` which belong to the `OMDoc` in Figure 1.

```
<definition id="def_order">
   <metadata>
      <Title xml:language="en">
        Definition of the order of a group element
      </Title>
      <extradata>
        <field use="mathematics"/>
        <abstractness level="neutral"/>
        <difficulty level="easy"/>
        <learning-context use="univ_first_cycle"/>
        <relation type="for">
          <ref theory="Th1" name="order"/>
        </relation>
        <relation type="depends_on">
          <ref theory="Th1" name="group"/>
        </relation>
      </extradata></metadata>
   <CMP xml:language="en" verbosity="3">
  ...  </CMP>
</definition>
```

Figure 1: Excerpt from an `OMDoc` for a definition

`field` describes the field to which the content of the item belongs. It enables ActiveMath to present items from particular fields (such as statistics, physics, or economy), if this is required by a pedagogical strategy. For instance, if students from different groups learn statistics, e.g., technicians, mathematicians, psychology students, they obtain different (motivating) examples and exercises from the appropriate field. The meaning of `abstractness` and `difficulty` is self-explaining. They serve to adapt the document to the learner's cognitive capabilities and learning progress. Currently, these metadata can have one of three different values. `learning-context` specifies for which learning context the material was intended originally. The possible values of

[2] `http://www.openmath.org/`

[3] `http://www.w3.org/Math/`

[4] For client-server applications, the annotations have to include the technical requirements to display or to invoke a material. They need to be known, in particular, to ensure that multimedia material is only offered, if the computer on the learners client can handle it.

[5] For a complete definition of the `OMDoc`-extensions see [2]

[6] such as the Learning Object Metadata (LOM) standardized by IEEE-LTSC (`http://ltsc.ieee.org/`)

[7] http://www.imsglobal.org/

[8] http://eml.ou.nl/

`learning-context` are those defined in the IMS-metadata standards. This information is important in case material from different sources is merged for a new course. Furthermore, the `OMDoc` in Figure 1 has a `verbosity`-attribute. The information about the verbosity of the textual parts allow the generation of different kinds of document such as slides and more verbose scripts.

The following metadata defined in the ActiveMath-DTD characterize exercises more precisely

- the targeted mastery-level of the exercise. Its values can be knowledge, comprehension, application, or transfer
- the task of the learner which can be calculate, check, explore, give_example, model, or prove
- level of interactivity and average learning time
- the technical type of the exercise. The ActiveMath-DTD allows for the values provide gap, mupad, maple, omega, multiple_choice, and fill_in currently.

ActiveMath employs these metadata to user-adaptively select exercises for a document and in the suggestion mechanism according to a particular teaching strategy that targets a particular mastery-level and adaptively supports skill acquisition.

The metadata `relation` is used to represent several relationships between `OMDoc` items. The type of this relation is specified in the `type`-attribute which can have the following meanings:

- the previous knowledge required to understand the item. For instance, the element in Figure 1 `depends-on` establishes dependencies on previous concepts.
- the similarity between the item and another one, e.g., for two examples, definitions, exercises etc. that are similar as, e.g., shown in Figure 2.
- a for-relationships which can be used in order to characterize an item with an additional functionality. For instance, an item that is a proof `for` a theorem could as well be an example `for` a method application or an example could be a counterexample `for` a concept.
- a citation-relationship referring to an additional `bib-extra` element which is defined in ActiveMath-DTD to enable a full featured citation mechanism.

One can argue that our `XML` specifications are "heavy on attributes". There are some pros and cons for this. Pros: when using attributes one can fix the default values for them, whereas the for body of an element we can only specify the type of data that can be placed inside (e.g. child elements, PCDATA etc.). Cons: no direct standards compliance (translation needed). Also note that, when using attributes, the need to interprete the labels is not introducing any additional effort for `XML` data manipulation engines.

The developement of ActiveMath metadata will be continued by: separating the metadata and the content databases; IMS content packaging (as soon as EML is integrated in IMS).

```
<definition id="def_leftcosets">
   <metadata>
      <Title xml:language="en">
        Definition of left cosets
      </Title>
      <extradata>
        <relation type="for">
          <ref theory="Th1" name="leftcosets"/>
        </relation>
           ...
        <relation type="similar">
          <ref xref="def_rightcosets"/>
        </relation>
      </extradata></metadata>
   <CMP xml:language="en">...</CMP>
</definition>
```

Figure 2: The relation to a similar definition

3. SUPPORT FOR AUTHORING

Authoring ontological `XML` is investigated in several projects, e.g., the SemanticWeb project[9] and the Ontology Editor Protégé [9].[10]

Authoring tools for the described truly semantic (mathematical) `XML` are still insufficiently developed. Such tools will not only have to support the author in employing or extending a (mathematical) ontology but also in authoring or choosing pedagogical metadata, in authoring exercises with external service systems, and support her in writing (abstracted) mathematical formulas that later can be presented via style sheets. This is a serious bottleneck currently.

So, what are the realistic alternatives currently? First, use a still preliminary tool, `QMath`, briefly described in §3.1 that supports the authoring of mathematical formulas. `QMath` provides at least some support but is not sufficiently comfortable for the average author. Second, translate from a syntactically oriented macro-based encoding and heuristically add semantics as described in §3.2. Third, wait until the open-source community including our group has produced a nice visual tool (some of it exists already, e.g., our visual editor for tables of content).

3.1 QMath

`QMath` is a tool and a migration format for producing `OMDoc` documents. It was developed by Alberto Gonzales Palomo and is currently used by ActiveMath authors to write `OMDocs`. A `QMath` document is easy to produce. This is partially due to the fact that `QMath` supports unicode and the author is free to write her formulas directly by using unicode symbols. Look, for instance at the following example, corresponding to the LaTeX source from the Figure 4:

```
 Definition:[<-df1]
:"The definition of cartesian product"
:for:[cartesian_product]
:depends_on:[def_pair]
...
$M × N Df suchthat(pair(x,y),x ∈ M & y ∈ N)$.
```

Figure 3: Excerpt from a `QMath` document

The author can define a context for a document. A context contains user-defined shortcuts for `OMDoc` elements. It also contains the references of symbols used in the document

[9] http://www.semanticweb.org/

[10] http://www.smi.stanford.edu/projects/protege/

to symbols defined in an OpenMath content dictionary. When the author uses a symbol, QMath suggests the symbol declaration procedure which assigns a meaning to the symbol by an explicit reference to a symbol in a content dictionary or via a pre-recorded context.

QMath supports metadata elements of OMDoc as well as some additional metadata of ActiveMath. This metadata support can be extended. As soon as a new element is declared in a document or in a context QMath performs a transformation to XML markup. [11]

3.2 LaTeX2OMDoc

For handling macro-based encodings, we face *two* demands: (1) the migration of existing mathematical learning document sources encoded in presentational languages, say LaTeX, and (2) the new encoding by authors/maths professors who are used to writing LaTeX and oppose authoring a truly semantic representation. In the following, we describe our efforts in both directions.

The migration of an existing large LaTeX document was the goal of a case study we conducted in 2001. The LaTeX sources of the document [4] were not intended to be migrated to a semantic representation originally. It unrestrictedly uses author-specific macros. Although the LaTeX source was already split into 'slices' and provided some dependencies, the document was designed pretty linearly rather than appropriate for a hypertext presentation and it was difficult to prepare for a reuse. For instance, it included text such as *"As we have seen in the previous example..."*.

The logical structure of the in the existing LaTeX document had to be carefully redesigned in order to obtain independently reusable items related via dependencies. For instance, many basic pieces in the text still included another one. E.g., introductions or elaborations contained a definition.

Another problem was the use of not-represented abbreviations. The text contained elements like ... *the correct notation for this should be* $\frac{\partial f}{\partial x}(a)$ *but we shall write only* $\frac{\partial f}{\partial x}$ *since it is clear that we are talking about the derivative in the point* a. A semantic representation would need to refer to the same mathematical object that may have different annotations, e.g., abbrev and a default.

The purely syntactic use of notation is one of the major problems. In a sentence like *Be carefull with our notations! In some cases* (a, b) *will mean an open interval and otherwise just an ordered pair.* (a, b) purely syntactical and its semantics is context-dependent. An automatic translation is almost impossible or at least has a highly context-dependent heuristics. The presentation-oriented representation and mis-use of LaTeX is another problem. This is obvious when the LaTeX encoding `$\{x: x\in A$ und $x$ ist rational $\}$`) of the formula $\{x : x \in A$ und x ist rational$\}$ is analyzed and shows that a mathematical formula is scattered into pieces and combined again by natural language text.

Apart the context-dependent and presentation-oriented representation, presentational and semantic information is mixed especially in the mathematical formulas and therefore heuristics for correctly parsing all formulas cannot be provided. This makes a fully automated translation practically impossible.

Our experience suggests that it is impossible to translate a LaTeX source automatically that has been written without the goal of a semantic representation in mind. This not only requires to implement many document-specific heuristics, it boils down to about 50% manual translation. Even worse, often the original encoding does not allow a unique translation to semantically represented formulas and may be author's work again.

A solution can only rely on a quasi-semantic markup in LaTeX that uses macros and environments to encode information needed for semantic knowledge representation and is extensible. This has been attempted in another case study, where we instructed 'conservative' authors to write strictly defined LaTeX sources and provided a tool for an automatic conversion to OMDoc via QMath.

We defined LaTeX macros and environments for the representation of semantic information and meta-data to support the automatic conversion to OMDoc. If an element has non-empty children, it is encoded by an environment, otherwise a macro is used[12]. We specified restrictions – in particular for writing formulas in that LaTeX. The most important restrictions are

- use the symbols already defined in OpenMath content dictionaries in order to ensure reusability,
- specify the interpretation of source formulas written in LaTeX, e.g. infix or prefix notation
- use the pre-defined LaTeX environments and macros for defining OMDoc elements, i.e., structure elements and metadata.

The following is an example of using special OMDoc-oriented LaTeX environments:

```
\begin{definition}{df1}{cartesian_product}
{The definition of cartesian product}
\depends-on{def_pair} ...
$M \times N \Df \suchthat{\pair{x}{y}}
{x \in M \and y \in N }$.
\end{definition}
```

Figure 4: Example of writing restricted LaTeX

- create a separate file to define new symbols and add XSL presentational information to the defined symbols, also define own DTD extension if necessary and XSL presentation for it.

If these requirement are met, our tool automatically converts the restricted LaTeX sources via QMath to OMDoc.

To summarize: as compared with a direct authoring of OMDoc in QMath, authoring in a restricted and augmented LaTeX is more familiar to mathematicians even if not strictly simpler. Although the direct control of the layout of a document by editing the generated PDF-document is very attractive to authors, it keeps the presentation and looses the representation and thus destroys the semantic and metadata information needed for the Semantic Web application.

[11] For more information on QMath see http://www.matracas.org/

[12] For more details see http://www.activemath.org/~ilo/articles/presentation2content112001.ps.gz

4. USAGE IN ActiveMath

Instead of a conclusion, we want to summarize what ActiveMath is able to do with the knowledge representation and what the future activities will be in this direction.

For the ActiveMath system, the reuse of content in different contexts is particularly important because its user-adaptivity implies that the same content can be presented in different ways depending on the user and in the learning situation.

ActiveMath' user-adaptive functionalities such as the presentation of the content and the dynamic suggestion generation use the structure information and metadata annotating the units of the content.

ActiveMath has the following components: a session manager, course generator, the mathematical knowledge base, a presentation planner, a user model, a pedagogical module and external mathematical systems (ActiveMath integrates several service systems for calculation, proof, and exploration such as the proof planner ΩMEGA [7] and the Computer Algebra Systems MuPAD [10] and Maple). Here, the user model is a component to store, read and update data about the learner's profile. It contains history of the actions, a list of preferences of the user and a list of competence assessments. The user's actions are analyzed by evaluators that calculate updates of the user model.

The course generation in ActiveMath is realized as follows: requests of the user are sent from the browser via a web-server to the session manager. When the user has chosen her goal concepts and scenario, the session manager sends this request to the course generator. The course generator contacts the mathematical knowledge base in order to calculate which mathematical concepts are required for understanding the goal concepts. Then the information about the user's knowledge is requested from the user model and the collected IDs of `OMDoc` items annotated with the user's knowledge mastery-levels are entered as facts into the knowledge base of the expert system. Then the rules are evaluated and generate an instructional list of `XML` items to be presented. Here, the metadata such as difficulty are not only used to select appropriate exercises and examples for a learner but also for the evaluation of the user's activities.

The `XML` content is transformed to a format suitable for presentation via `XSL`. An `XSL` style sheet specifies the presentation of our `XML` documents, by describing how an instance is transformed into `HTML` or to LaTeX or Flash.

The semantic representation is a basis for merging content from different sources and presenting the merged content consistently.

The ActiveMath knowledge representation is providing two ontologies: the mathematical and the educational one. Mathematical ontology is also useful for other math applications. The mathematical concepts (elements of ontology) are the skeleton (macro level) of a document, and the educational ones provide the information for building the micro level structure.

Alternative Usages

ActiveMath' support for exploratory and interactive learning will be improved. This includes the investigation of elaborate search functions based on the the semantic and partially formal representation.

A next step is the machine-understandable description of mathematical operations. Such descriptions are useful in many situations, including the automated advise to a user for choosing an appropriate system to perform a task or for agent-based computations (see [5]).

Semantically represented repositories will be useful not just for learning environments but also for working mathematicians. For instance, `OMDoc`-structured repositories can improve the organization and searchability of mathematical knowledge. Today the digital libraries,have to face the manually controlled entry of author and classification information and often make use of keywords authored by reviewers. Today the search capabilities are limited to textual and keyword search.[13]

We understand our research as part of the larger European initiative for web-based mathematical knowledge representation and management. Its first workshop [1] covered topics ranging from publishing of large collections of electronic preprints to tools for managing mathematical documents. There is hope for a critical mass of content encoded in a semantic `XML` since the initiative will work on this as well as on tools to maintain and use the content data and metadata.

5. REFERENCES

[1] *Electronic Proceedings of the First International Workshop on Mathematical Knowledge Management*, September 2001.

[2] J. Büdenbender, G.Goguadze, P.Libbrecht, E.Melis, and C.Ullrich. Metadata for ActiveMath. Seki Report SR-02-01, Fachbereich 14 Informatik, Universität des Saarlandes, 2002.

[3] O. Caprotti and A. M. Cohen. Draft of the open math standard. Open Math Consortium,1998, `http://www.nag.co.uk/projects/OpenMath/omstd/`.

[4] B.I. Dahn and H. Wolters. *Analysis Individuell.* Springer-Verlag, 2000.

[5] M. Dewar and D. Carlisle. Mathematical software: the next generation? In [1]

[6] M. Kohlhase. `OMDoc`: Towards an internet standard for the administration, distribution and teaching of mathematical knowledge. In *AISC'2000*, 2000.

[7] E. Melis and J.H. Siekmann. Knowledge-based proof planning. *Artificial Intelligence*, 115(1):65–105, 1999.

[8] E. Melis, E. Andres, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. ActiveMath: System description. In Johanna D. Moore, Carol Redfield, and W. Lewis Johnson, eds, *Artificial Intelligence in Education*, pages 580–582. IOS Press, 2001.

[9] N. F. Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of Protégé-2000: Combining interoperability and flexibility. In *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, 2000.

[10] A. Sorgatz and R. Hillebrand. MuPAD - Ein Computeralgebra System I. *Linux Magazin*,(12),1995.

[11] The Dublin Core Metadata Initiative, 1998, `http://purl.org/DC/`.

[13] The American Mathematical Society's E-Print service, e.g., allows the search through the reviews only which are traditionally encoded in the TeX language. Searching through formulas in this language is one of the most unpredictable tasks as formulas are encoded for presentation only.