Context-Aware Adaptation

A Case Study on Mathematical Notations

Christine Müller
c.mueller@jacobs-university.de
Department of Computer Science
University of Auckland
Private Bag 92019, Auckland, New Zealand

Michael Kohlhase
m.kohlhase@jacobs-university.de
Computer Science Department
Jacobs University Bremen
Campus Ring 1, Bremen, Germany

Michael Kohlhase is Professor for Computer Science at Jacobs University, Bremen and Vice Director at the German Research Centre for Artificial Intelligence (DFKI) Bremen. His current research interests include Automated Theorem Proving and Knowledge Representation for Mathematics, inference-based techniques for Natural Language Processing, and computer-supported Education. He has pursued these interests during extended visits to Carnegie Mellon University, SRI International, Rice University, and the Universities of Amsterdam and Edinburgh.

Christine Müller is a PhD Candidate at the Jacobs University. In collaboration with Cristian Calude, University of Auckland, NZ, she is conducting a case study on the mathematical community and their practices with an emphasis on proof explanations. Her current research focuses on *active documents* that provide dynamic personalized views on technical material. She is coordinator of the *panta rhei* project and the Special IG "*Scientific Communities of Practice*" of the Joining Educational Mathematics Network.

Abstract

In the last two decades, the World Wide Web has become the universal information source. Search engines can efficiently serve daily information needs due to the enormous redundancy of relevant resources on the web. For educational and scientific information needs, the web functions much less efficiently: Scientific publishing is built on a culture of unique reference publications, and moreover documents abound with specialized structures such as technical nomenclature, notational conventions, references, tables, or graphs. Many of these structures are peculiar to specialized communities determined by nationality, research group membership, or adherence to a special school of thought. To keep the much-lamented "digital divide" from becoming a "cultural divide", we have to make online material more accessible and adaptable to individual users.

In this paper we attack this goal for the field of mathematics where knowledge is abstract, highly structured, and extraordinarily interlinked. Modern, content-based representation formats like OpenMath or content MathML allow us to capture, model, relate, and represent mathematical knowledge objects and thus make them context-aware and machine-adaptable to the respective user contexts. Building on previous work, which can make mathematical notations *adaptable*, we employ user modeling techniques to make them *adaptive* to relieve the reader of configuration tasks. We present a comprehensive framework for adaptive notation management and evaluate it on the proof-of-concept prototype *panta rhei*.

Keywords. User Modeling, Practice-Oriented Hypermedia Adaptation, Mathematical Knowledge Management

1 Introduction

Mathematics is one of the oldest disciplines and a basis for most modern science. Looking at our scientific history, many innovations originate in mathematics: Mathematicians have paved the way to new scientific inventions, allowing other science to develop mathematical methods further and to provide a practical use. For example, in 1854 the English mathematician *George Boole* wrote his book on what we now call "Boolean Logic" (Boole, 1858), providing the basis for *computer science*. More recent examples come from the area of *cryptography*, as modern algorithms such as RSA (Rivest, Shamir, and Adleman, 1978) which eventually allowed to develop secure and stable methods for encrypting and decrypting numbers, could not have been developed without number theory. Looking at *image compression*, first algorithms only allowed a compression up to 50%. Nowadays, JPEG provides a compression of 95% based on *Discrete Fourier Transform (DFT)*. Fourier Transform (or Fourier analysis) is named after the French mathematician *Jean Baptiste Joseph Fourier* (1768-1830), who contributed the first mathematical discoveries and insights into the practical usefulness of the underlying mathematical techniques. We could go on with examples that illustrate the importance of mathematics. However, we believe that the most important contribution of mathematics is the *scientific language* that it provides to all other disciplines.

Looking at the *history of the mathematical language*, we observe a tendency towards increased rigour: Early publication include more natural language as many notations had not yet been developed. For example, around 250 AD the Greek mathematician *Diophantus* could not draw on symbols such as $=$ for *equality*, $<$ and $>$ for *less/greater than*, as well as $\leq$ and $\geq$ for *less/greater or equal than*, since during his productive period *number theory* did not yet provide these notations. Throughout the years, more and more mathematical symbols and notations have been added. Nowadays, mathematical language is a mixture of highly specialized notations and natural language, where notations make up 30-60% depending on the type of mathematical text. Mathematical notations have become an essential part of mathematical language, similar to musical notation systems, which are fundamental for the creation and communication of compositions. *Modern science* is *inconceivable* without a precise notation system: Notations *ease communication* of all mathematical practitioners as they *reify* mathematical ideas into compact and precise forms, which, conversely, have to be *interpreted* by the recipients. Mathematics is often said to be "the language of science". Gallileo Gallileo even went so far as to say "Mathematics is the language in which God has written the universe." (Galilei, 1623).

Nevertheless, the increased formalization and need for interpretation of mathematical language also bears its challenges: Mathematical notations can *complicate communication* and *acquisition processes*, in particular, for less experienced consumers. This is due to the fact, that mathematical notations are *context-dependent* and *vary strongly* among different communities and individuals (see Section 2.1 for examples). Consequently, notations can cause *ambiguities* and *misunderstanding* and, thus, may *hamper learners* and *collaborations*. Even though notations are an essential part of mathematical texts, we are still not able to fully control them, e.g. to adapt them to a reader's background and preferences.

We observe another trend, which provides the bases for an *automatized management of mathematical notations*: More and more scientists have *changed* their *mathematical practice* and opened up towards modern technologies that allow them to find and share mathematical results more easily and to even automatize computation, verification, and reasoning tasks. As a consequence, mathematical knowledge is produced and applied at an unprecedented rate (Odlyzko, 1995) calling for more advanced support for managing mathematical knowledge. Recently, scientists of interdisciplinary areas have formed the new scientific field of *mathematical knowledge management* (MKM) (MKM), which aims at developing better ways to articulate, organize, disseminate, and access mathematical knowledge. In particular, they share a common vision of *stepwise formalizing all mathematical knowledge* and of making it *available on the World Wide Web*. In this context, more and more research contribute to an improved management of mathematical notations. Two XML-based representation formats have been specified to provide a better display and machine-readable access of mathematical notations on the World Wide Web, i.e. MathML (W3C, 2003) and OpenMath (OpenMath). Moreover, workflows have been developed that allow to generate either format (W3C, 2003; OpenMath) from various scientific editors: The LaTeXML (Miller, 2007) allows to generate the respective XML from LaTeX. Editors of several web-based (authoring and E-Learning) environments, such as Sentido (González Palomo, 2006), MathDox (Cuypers, Cohen, Knopper, Verrijzer, and Spanbroek, 2008), ActiveMath (Melis and Siekmann, 2004), ConneXions (Henry, Baraniuk, and Kelty, 2003), and SWiM (Lange, 2008), produce MathML and OpenMath, respectively. In addition, several mathematical computation systems provide an import & export from and to the two standards.

However, services based on mathematical notations are still limited. The search engine MathWebSearch (MathWebSearch) facilitates to search mathematical formula in MathML and OpenMath. The MathPlayer (Mathplayer) plugin for Internet Explorer can read out MathML-encoded mathematical formulae for the sight-impaired and the mathematical braille translator (Archambault, Stöger, Fitzpatrick, and Miesenberger, 2007) outputs MathML for braille devices. Unfortunately, there is little consideration for the context and adaptation of mathematical notations to facilitate understandings, sharing of material, as well as online collaborations.

(Smirnova and Watt, 2006; Naylor and Watt, 2001) point to different *notation contexts* that can cause multiple notations of the same mathematical concept, namely *area of application*, *national conventions*, *level of sophistication*, the *mathematical context*, and the *historical period*. They also provide the first approach towards *modeling notation preference*: The author provide a *notation selector* (Smirnova and Watt, 2006) that allows users to design *user-specific XSLT stylesheets* for the conversion of mathematical notations based on (W3C, 2003; OpenMath). Apart from (Smirnova and Watt, 2006; Naylor and Watt, 2001), only the ActiveMath group (ActiveMath) has started to address mathematical notation representation and modeling: The E-Learning systems provides an adaptive selection of examples and sequencing of learning objects to generate user-specific courses, but does not yet provide the adaptation of notations towards the notation preferences and background of single users.

This paper introduces a *context-aware and adaptive framework for managing mathematical notations* as we believe adaptation to be an essential service for the distribution, sharing, and understanding of mathematical web material. In Section 2 we briefly introduce mathematical notations as well as their markup and describe our previous presentation framework. Section 3 introduces the components of the user model and our user modeling approach as well as the creation, activation, and maintenance of user models. Section 4 illustrates the exploitation of the user model for user-specific adaptation of mathematical notations and sketches the revision of our previous conversion algorithm. In Section 5, we present the system-independent representation of the user model as well as our generic adaptation components. We further describe our prototype application and provide an evaluation based on informal interviews of mathematical researchers and lecturers. Related work and conclusion are provided in Section 6 and Section 7.

2 Mathematical Notations and Adaptation

2.1 Mathematical Notations & Notation Systems

Mathematical notations denote mathematical concepts, i. e., the objects we talk and write about when we do mathematics: Rather simple objects like numbers, functions, triangles, matrices, and more complex ones such as vector spaces and infinite series. It is crucial to note that mathematical notations are *highly interdependent*; in mathematics we speak of *notation systems*, i. e. collections of notations that work well together. Consequently, the choice of a specific notation for a concept requires to use notations from the same system for all other concepts. For example, if we look at the notation for *subset* and *proper subset*, we can use $\subseteq$ and $\subset$ versus $\subset$ and $\subsetneq$. In the first combination, $\subset$ denotes the *proper subset*, while in the second combination it denotes *subset*. Consequently, when adapting the notation of subset from $\subseteq$ to $\subset$, we need to also change the notation for proper subset from $\subset$ to $\subsetneq$. Otherwise, we end up with the same notation for two different mathematical concepts, which eventually destroys the discern able meaning of the mathematical formula.

Mathematical Communities and their Notations We observe *mathematical communities*, which prefer different *notation systems*: For example, if we look at a *Russian* and *Western* mathematical journal we will find that two different notation systems are used. Partly, the notations between Russian and Western researchers differ as they build on different concepts. However, also the *overlapping concepts* used by both groups are denoted with very different sets of notations. Moreover, even if Western researchers used and defined concepts solely used by Russians, they would denote them very differently staying conform to the type of notations in their systems.

Mathematical areas are further divided into different *schools* that originally evolved based on individual styles of influential mathematicians. For example, (Chaitin, 1987) and (Li and Vitanyi, 1997) use different notations to denote the same concepts (plain and prefix free complexity) in the field of *algorithmic information theory* (AIT): Chatin/Calude use $K(x)$ and $H(x)$, while Li/Vitany use $C(x)$ and $K(x)$.

Individual Styles We can also observe that individual author styles differ within schools or communities. For example, some mathematical authors are more *formal*, while others prefer to include more *natural language terms*. For example, consider the mathematical statement "Let n equal 2 times m square. Choose a natural number k so that k is less than or equal to n." in contrast to the more compact and formal "Let $n = 2m^2$. Choose a number $k \leq n$.". Some mathematicians feel that the latter is more *easier to read*, while others reject it, as they believe that symbols should not be part of the *prose text*. Consequently, mathematicians tend to prefer a specific *notation system* and *style* and often *reject* material with notations that differ to their own. In particular, different notation systems cause problems for the integration of (online) course materials from different authors as they cause inconsistencies and a tedious refactoring of the combined material.

2.2 Representation Mathematical Notations

To provide automated services such as the adaptation of mathematical notations, we represent mathematical objects in formats like MathML (W3C, 2003) a W3C recommended format for high-quality presentation of mathematical formulas on the Web or OpenMath (OpenMath), a content-oriented format that concentrates on the meaning of objects. In fact MathML has a sub-language that is equivalent to OpenMath, but we will concentrate on the presentational functionality of MathML for simplicity. Similarly, we will use OpenMath synonymously with "content markup".

OpenMath Representation

```
<om:OMOBJ>
  <om:OMA>
    <om:OMS cd=”combinat1”
                name=”binomial”/>
    <om:OMV name=”n”/>
    <om:OMV name=”k”/>
  </om:OMA>
</om:OMOBJ>
```

MathML Representation

```
<m:mrow>
  <m:mo fence=”true”>(</m:mo>
  <m:mfrac linethickness=”0”>
    <m:mi>n</m:mi>
    <m:mi>k</m:mi>
  </m:mfrac>
  <m:mo fence=”true”>)</m:mo>
</m:mrow>
```

Presentation

$$\binom{n}{k}$$

Figure 1: OpenMath and MathML representation of the binomial coefficient.

Figure 1 provides the OpenMath and MathML representations of $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, the number of k-element subsets of an n-element set. The OpenMath expression on the left captures the functional structure of the expression by representing it as the application (using the `OMA` element) of the "binomial coefficient" function (represented by an `OMS` element) applied to two variables (`OMV`). Note that the `cd` and `name` attributes characterize the binomial function by pointing to a definition in a **content dictionary** (CD) (OMCD-Core), a specialized document that specifies *commonly agreed* definitions of basic mathematical objects and allows machines to distinguish the meaning of included mathematical objects. In contrast to this, the presentation MathML expression in the middle marks up the appearance of the formal when displayed visually (or read out aloud for vision-impaired readers): The formula is represented as a horizontal row (`mrow`) of two stretchy bracket operators (`mo`) with a special layout for fractions (`mfrac`, where the line is made invisible by giving it zero thickness) where the numerator and denominator are mathematical identifiers (`mi`). The aim and strengths of the two formats are complementary: OpenMath expressions are well-suited for information retrieval by functional structure and computation services, while MathML is used for display: MathML-aware browsers will present the middle expression in Figure 1 as $\binom{n}{k}$.

```
<m:semantics>
  <m:mrow id="top">
    <m:mo>(</m:mo>
    <m:mfrac linethickness="0">
      <m:mi id="left">n</m:mi>
      <m:mi id="right>k</m:mi>
    </m:mfrac>
    <m:mo>)</m:mo>
  </m:mrow>
  ...
  <m:annotation−xml>
    <om:OMOBJ>
      <om:OMA xref="top">
        <om:OMS cd="combinat1"
                name="binomial" />
        <om:OMV name="n" xref="left"/>
        <om:OMV name="k" xref="right"/>
      </om:OMA>
    </om:OMOBJ>
  </m:annotation−xml>
</m:semantics>
```

Figure 2: Parallel Markup: Combining OpenMath and MathML

In order to combine both markup aspects, MathML allows *parallel markup* (W3C, 2003) with fine-grained cross-references of corresponding sub-expressions. Figure 2 provides the parallel markup for the example in Figure 1: The `semantics` element embeds a presentation MathML expression and an `annotation-xml` with the respective OpenMath expression. The `id` and `xref` attributes specify corresponding subterms. An application of this would be that a user can select a subterm in the presentation MathML rendered in a browser, so that a context menu option could send the corresponding OpenMath sub-expression to e. g. a computer algebra system for evaluation, simplification or graphing.

In order to automatically adapt mathematical notations, we need to be able to vary the displayed presentation MathML. This is usually done by parameterizing the process by which presentations are generated from a given content representation. In our approach, we represent the *mappings* between an OpenMath expression and all alternative MathML representations declaratively. Conceptually, these mappings represent *mathematical notation practices* as they explicate the *choice of mathematical notations* of the user. In order to make adaptation *practice-* and *context-aware*, we need to provide a conversion workflow that takes notation practices and concrete context as input and adapts the respective notation for the user. Before we present our framework, we will review the state of the art.

2.3 Representing Notation Practices

We will recapitulate our representation framework for mathematical *notation practices* that supports a flexible and *context-aware* presentation process (Kohlhase, Müller, and Rabe, 2008) to make our exposition self-contained: We reify *notation practices* into *notation definitions*, which used as parameters in the conversion from OpenMath to MathML (or parallel markup, of course).

```
<notation xmlns="http://omdoc.org/ns"
          xmlns:m="http://www.w3.org/1998/Math/MathML"
          xmlns:om="http://www.openmath.org/OpenMath">
  <prototype>
    <om:OMA>
      <om:OMS cd="combinat1" name="binomial" />
      <expr name="arg1"/>
      <expr name="arg2"/>
    </om:OMA>
  </prototype>
  <rendering context="hasLanguage:Russian,ru">
    <m:msubsup>
    <m:mi>C</m:mi>
    <render name="arg1"/>
    <render name="arg2"/>
    </m:msubsup>
  </rendering>
  ...
  <rendering context="hasLanguage:German,de">
    <m:mrow>
      <m:mo>(</m:mo>
      <m:mfrac linethickness="0">
        <render name="arg1"/>
        <render name="arg2"/>
      </m:mfrac>
      <m:mo>)</m:mo>
    </m:mrow>
  </rendering>
  <rendering context="hasLanguage:French,fr">
    <m:msubsup>
      <m:mi>C</m:mi>
      <render name="arg2"/>
      <render name="arg1"/>
    </m:msubsup>
  </rendering>
</notation>
```

Figure 3: XML Representation of a Notation Practice.

Figure 3 shows a notation definition for binomial coefficients. The `prototype` pattern matches OpenMath expressions such as the one in Figure 1 instantiating the named `expr` elements. The concrete presentation for the expression is induced from the `rendering` elements recursively rendering the instances `render` elements corresponding by name. Note that there can be multiple `rendering`s for a pattern, which are distinguished by the `context` attribute, which associates them with specific context parameters. In the example, the nationality of the respective notations are added. This allows to distinguish the

German, Russian, and French notation of the binomial coefficient. Analogously, further context parameters such as the expertise level (novice, intermediate, expert) or area of application (mathematics, physics) can be added.

2.4 Adaptability of Mathematical Notations

In (Kohlhase et al., 2008) we proposed a context-aware conversion algorithm for selecting appropriate presentations: First we collect all notation definitions for a mathematical object, then we collect the user's context parameters for the conversion, and finally we select an appropriate `rendering` element which best fits to the current context and apply it to generate a presentation for the mathematical object. To provide a flexible and context-aware conversion algorithm, we provide various options to collect notation definitions as well as concrete context parameters (see (Kohlhase et al., 2008; Kohlhase, Lange, Müller, Müller, and Rabe, 2009) for details). Given the notation definition in Figure 3 and a concrete context parameter, the mathematical object in Figure 1 can be presented differently: For example, depending on the nationality selected by the user, the binomial coefficient is presented with its German $\binom{n}{k}$, Russian C^n_k, or French notation C^k_n.

3 Modeling Mathematical Notation Preferences

Note that the conversion algorithm introduced above (see Section 2.3 and 2.4) allows experienced users — e.g. authors that are familiar with the proposed markup of notations and are skilled in selecting the appropriate options — to be in control of the adaptation. For readers, however, who do not want to invest into insights into the semantics of the document but rather consume the presented material efficiently and conveniently our approach expects too much extra effort. Therefore we need to provide an adaptive workflow that requires little user effort no knowledge of the underlying representation. Our previous workflow required *user-driven* personalization and modifications and cannot not offer automatized *system-driven* personalization based on a model of the user's notation preferences and contexts. Conceptually, it was a *adaptable* representation, where we need an *adaptive* framework for the reader. In the further course of this section we extend our notation framework with user modeling techniques, thus transforming it from an adaptable to an *adaptive notation management*.

3.1 Components of the User Model

We integrate user models into our notation framework to represents the user's notation background and preferences, but have observed that notation preferences highly depend on the user's current situation. For example, in LaTeX the author can choose between a *display* (for presentations) or *text mode* (for textbooks) depending on the respective format. However, the selection is static and cannot be adapted based on given context parameters. Moreover, a mathematician might try to use simpler notations in a lecture than in a talk for her research community. However, currently she is not supported with a respective notation management in her working environment. Or consider a mobile scenario: More and more researchers use mobile devices to get work done on the road, thus adaptation of mathematics for different devices becomes an issue. Having made these observations, we take on the work of (Nauerz, Welsch, Bakalov, and König-Ries, 2008), who use context models in addition to a static user model, and adapt their approach to model notation preferences. Figure 4 presents the three constituents of our *user model*: A *static profile*, a *default notation model* and a *set of situational notation models*.

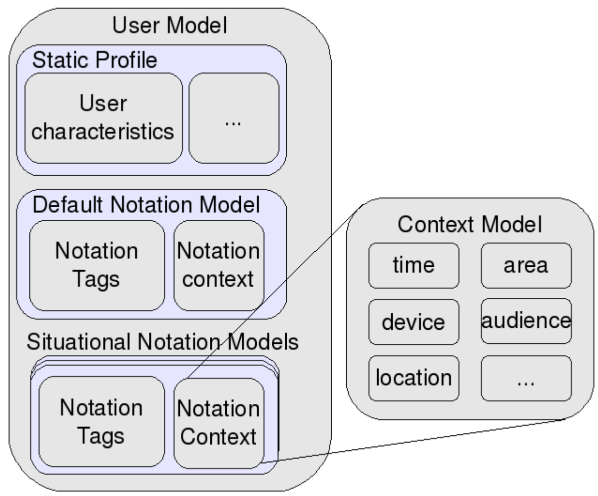

Figure 4: Constituents of the User Model

The static profile represents static user characteristics such as the age or native language of the user. In this paper we focus on modeling notation preferences, but aim towards an extensible model, which can be enriched with further user information such as her *background and interest*, *tagging behavior*, or *social network* (cf. (Nauerz et al., 2008)).

The default notation model provides the general notation background of a user and is used as *fallback* for the notation adaptation if no concrete situational model applies. It consists of (1) a *default notation context* representing the user's general notation background based on a set of *context parameters* and (2) *default notation tags* representing the user's general notation behavior, i.e., which notation she uses, knows, prefers, or dislikes.

Situational notation models represent the user's notation background regarding a concrete situation: For example, we can provide different models to represent the user's notation behavior during her *introductory computer science lecture*, a *conference talk in her research community*, or her *private studies at home*. Situation models consist of (1) a *situational notation context* representing the user's current situation based on a set of *context parameters* and (2) *situational notation tags*, i.e., a set of notation tags used within the concrete situation.

Notation contexts consist of a set of *context-parameters*, i.e., *dimension-value pairs* where the *Dimensions* provide *classes of context descriptions* such as `hasDate`, `hasTime`, `hasLocation`, `hasDevice`, `hasTask`, `hasArea`, `hasAudience`, `hasLanguage`, `hasEvent`, or `hasLayout`. *Context values* instantiate dimensions with concrete entries such as `2008-09-24`, 14:12:00, `Athens`, `desktop`, `talk`, `mathematics`, `information systems & psychology`, `English`, `WSKS-Conference`, or `display-mode`. Note that the *notation context* is limited to parameters that we find relevant for the modeling of notation preferences. However, we aim towards a general context model such as in (Nauerz et al., 2008).

Notation tags are *weighted* and *annotated* references to notations. The *weights* express how often and recent a user interacts with a specific notation. This interaction is either direct by choosing a specific notation or indirect by interacting with a page with respective notations. Optionally, notation tags can be associated with a *status*, which expresses whether the referenced notation is used, preferred, disliked, or known. Moreover, context-parameters as well as references to mathematical objects, i.e., OpenMath expressions, can be added (see more details in (Kohlhase et al., 2009)).

3.2 Implicit and Explicit User Modeling

We can take two alternative approaches to extract and collect information for our user models: An implicit and explicit modeling (Brusilovsky, Kobsa, and Neidl, 2007). **Implicit modeling** measures the users interest and preference by exploiting the user's activities in the system. Assumptions and inferences on the user behavior can be made without requiring extra efforts of the user. We apply two techniques of the implicit approach: *Web mining* and *tagging/ rating behavior analysis.*

Web mining (Liu, 2007) applies *data mining techniques* to discover (usage)-patterns within web data. It aims at extracting usage patterns from *log data* to model certain aspects of the behavior of users or communities. Analyzing logs allows us to extract information about how often user interact with certain notations. For example, we can identify which notations are *explicitly changed* by the user (see Section 4) and assume that this expresses her preference of the notation. Alternative, based on her *target hits* we can observe the user's interaction with certain pages allowing us to track her interaction with the included notations. We assume that if a user accesses a page often, her familiarity with the included notations increases. Explicit interactions with notations are considered more influential than interactions with pages. We thus *measure the directness* of the interaction, so that the explicit change of notations has a higher impact on the user model than reading a page (and its included notations). In addition, we consider more recent interactions more important and thus also apply a *time-weighting factor*. The collected information are used to add *weighted annotated notation tags* to the user model, or rather the sub-model that is currently active (see Section 3.3): The weights of the tags are computed based on the directness of the interaction and the time-weighting factor. The annotations express user perspective on the notation: We add `status:used` for explicit changed notation and `status:seen` for notation inside pages to the notation tags.

Tagging/ rating behavior analysis (Nauerz et al., 2008) allows analyzing a user's tags of a page to infer her interest in the *included* notations, while her rating expresses her personal opinion on the quality of the page and thus can be interpreted as an explicit approval or disapproval with the included notations. Tags and ratings can be used to create weighted annotated notation tags, where the weight express the frequency and recentness of a tag/rating and the annotation expresses the user's opinion: We add `status:seen` for tags and `status:likes` or `status:dislikes` for the respective rating value to the notation tags.

But note that users interact with *presented* objects rather than their notations. To adapt these objects multiple notation definitions are required. For example, if we consider the expression $a + b - c$ we have to apply two notation definitions, one for the symbol `plus` and `minus`, to convert it into an alternative presentation. Our approach preserves *references* of the presented object to the respective notation definitions that were applied (see Section 4 for more details). Consequently, we can track the considered notation definitions for each interaction, thus, allowing us to implicitly model all notation tags, which references the required notation definitions and carry a reference to the respective object, e.g., $a + b - c$.

In contrast to the implicit modeling, **explicit modeling** requires direct modification of the user model by the respective user, i.e., users have to invest extra effort and explicitly enter data to either the default or situational models.

Questionnaires Users can explicitly enter context parameters to describe there interest and background. We provide questionnaires that allow users to specify their general background, e.g., the user's native language, location, or intended audience of a talk. The context parameters are then added to the respective notation context.

3.3 Creation, Activation, and Maintenance of User Models

The user modeling approaches in Section 3.2 require *prior interactions* and *explicit inputs* of users before being useful for the adaptation of notation. In this section, we address a problem that many software systems face: Even though more and more systems allow users to specify interaction preference or even employ user modeling techniques, many systems are islands with this respect. In particular, different systems cannot share user models or predict in the absence of prior interactions. Consequently, users cannot reuse their user models in other systems or even initialize their models based on other users' settings. Below we provide details on the *creation and activation* of our user models as well as the *reuse* and *transfer* of models across systems.

Creation & Activation Analogously to the context models in (Nauerz et al., 2008), situational models are manually created by users to specify different notation settings: For example, a user might choose to create a different notation model for her *introductory computer science lecture*, a *conference talk in her research community*, or her *private studies at home*. Users can manually switch between their situational models and default model or can allow the system to switch automatically. While a situational model is *active*, all identified context-parameters and notation tags are stored in the model. If no situational model is active, the entries are added to the default notation model.

Sharing We build on existing user modeling ontologies, reusing their *domain-independent ontological concepts*, while staying compatible with their definitions. We provide tools that allow to convert our user models in respective representation standards such as RDF and

OWL. Consequently, our user models can be interpreted and reused in other adaptive systems. Vice versa, prior interactions and settings from other systems can be imported into our models, thus, reducing the required initial effort from users. Moreover, user and community models no longer have to be maintained inside specific systems. Instead, we can provide a central facility for maintaining them (see more details on the representation of user models in Section 5.1).

Central Model Maintenance In (Müller and Kohlhase, 2008a) we proposed a toolkit, which, similar to user modeling servers or services (see Section 6), outsource the user modeling maintenance from user-adaptive application systems into a central entity. (Müller and Kohlhase, 2008a) provides a scenario for integration the toolkit with a mathematical E-Learning system: When logging into the system for the first time, users can download their user model or the user model of a friend to initialize the system's setting. Moreover, the toolkit provides functionality to create community models from user models (see (Müller, 2008)) and to initialize user models from community models.

4 Exploiting User Models for Automatic Adaptation

In this section we provide the components of our notation framework and provide details on the revised conversion algorithm. We claim that usability of mathematical systems can be improved by automated adaptation and have started with an adaptive context-aware conversion of mathematical notations.

The specifics of any adaptivity mechanism depend on the underlying *content representation format*. We base our work on our Open Mathematical Document Format (OMDoc) (Kohlhase, 2006) which provides markup in *different layers*: The OMDoc format extends OpenMath and MathML (for the formula layer) with markup primitives for the structure and interrelations of mathematical objects expressed as mathematical statements, e.g., definitions, theorems, and proofs. OMDoc allows to represent content dictionaries

(CD), which explicitly represent context for mathematical symbols and formulae, as OMDoc documents containing mathematical statements. A very expressive infrastructure for *inter-CD relations* facilitates concept inheritance, parametric reuse, and multiple views on mathematical objects and statements. The links between theories represents prerequisites of knowledge objects and, thus, contribute to a more semantic specification of *constraints and goals*. On the *presentation layer*, OMDoc provides tools for the conversion into several formats, such as PDF or XHTML and provides a sophisticated presentation-pipeline for the rendering of mathematical notations.

However, our notation framework is not limited to materials in OMDoc but provides basic support for any XML-based content representation (including XHTML), which build on the MathML and OpenMath standards and provide parallel markup. Consequently, formats such as CNXML (Hendricks and Galvan, 2007) of the Connexions system (Henry et al., 2003) or potentially InkML (Chee, Franke, Froumentin, Madhvanath, Magana, Russell, Seni, Tremblay, Watt, and Yaeger, 2006) for pen-based computing can be supported (with reduced functionality). Note that even though we focus on adaptive notations in this paper, our framework can be expanded for further adaptations, such of the *selection* or *sequencing* of content fragments to for a *coherent user-specific* reading experience.

4.1 Adaptive Notation Framework

Figure 5 provides the architecture of our extended notation framework, which can be used by any **(mathematical) system** that contains **semantic content** *structured* via the **OMDoc model**, which provides markup on four different layers: The **object**, **statement**, **theory**, and **document** layer (see (Kohlhase, 2006) for more details). For the *adaptation* we need information about individual users and their notation preferences, thus we maintain a **user model**. We argue that notations critically depend on the user's situation and thus include **situational models** in addition to the **default notation model** (see Section 3.1). These models are initialized based on different *implicit and explicit modeling*

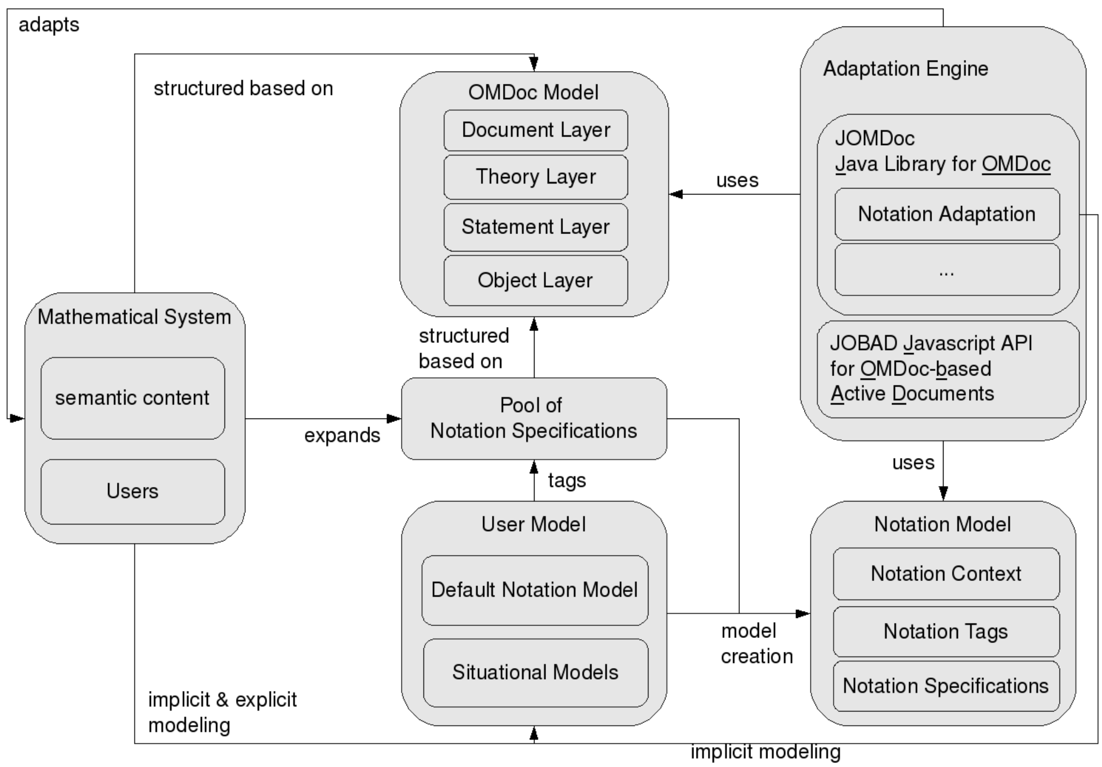

Figure 5: Extended Notation Framework, on the basis of (Nauerz et al., 2008)

techniques (see Section 3.2). During the modeling, users can provide their individual notation definitions, which *expand* a central **pool of notation definitions**, which is *shared* among several mathematical systems and *structured* according to the OMDoc model. The notations inside this pool are referenced by *notation tags* in the user models and provide the individual perspectives on the pool. These tags allow to explicitly select notations that the users prefer (and accordingly to avoid notations that they dislikes). In addition, user models include context parameters, i.e., the user's **notation context**, which allow to intensional select appropriate notations from the pool: For example, the parameter `language:German` prioritizes German notations in the adaptation. Notation tags and context as well as a selection of notation definitions *form* the **notation model**. Based on the OMDoc and notation model, an **adaptation engine** *triggers* the *adaptation* of material in the mathematical system drawing on two central components: **JOMDoc** (JOMDoc) and **JOBAD** (JOBAD) (see Section 5.2).

5 Implementation & Case Study

5.1 System-Independent Representation of User Models

We use an *ontological* representation for our user models. In contrast, to *non-ontological* representation formats, such as sets of relational database tables or XML files, ontological representation formats are not limited to *describing* user characteristics but support *automated reasoning* about the user model's content. Moreover, building on semantic web standards for the representation of ontologies, such as RDF (RDF) and OWL (Euzenat and Patel-Schneider, 2003), allows as to *share* user characteristics across a range of systems, allowing us to make use of previously initialized data for our adaptations. The user modeling community focuses on ontology based approaches and provides several reusable user model ontologies (cf. (Andrejko, Barla, and Bielikova, 2007)). We reuse *domain-independent ontological concepts* of existing specifications, such as the General User Model Ontology GUMO (Heckmann, Schwartz, Brandherm, Schmitz, and von Wilamowitz-Moellendorff, 2005), and add representations for mathematical characteristics to model (notation) preferences and contexts while staying compatible with existing ontologies.

```
<!DOCTYPE rdf:RDF
[<!ENTITY u ”http://omdoc.org/user#” > ]>
<rdf:RDF
  xmlns      =”&u;”
  xmlns:rdf=”http://www.w3.org/...#” >
<owl:Ontology rdf:about=””>
 ...
<owl:Class rdf:ID=”Area”>...</owl:Class>
<owl:Class rdf:ID=”Mathematics”>
  <rdfs:subClassOf rdf:resource=”&u;Area” />
</owl:Class>
<owl:Class rdf:ID=”Language”>...</owl:Class>
 ...
<rdf:Description rdf:about=”&u;hasLanguage”>
  <rdfs:domain rdf:resource=”&u;User”/>
  <rdfs:range rdf:resource=”&u;Language”/>
</rdf:Description>
<rdf:Description rdf:about=”&u;hasArea”>
  <rdfs:domain rdf:resource=”&u;User”/>
  <rdfs:range rdf:resource=”&u;Area”/>
</rdf:Description>
 ...
</owl:Ontology rdf:about=””>
</rdf:RDF>
```

```
<!DOCTYPE rdf:RDF
[<!ENTITY u ”http://omdoc.org/user#” > ]>
<rdf:RDF
  xmlns =”http://omdoc.org/user#”
  xmlns:rdf=”http://www.w3.org/...#” >
 ...
<rdf:Description
  rdf:about=”http://cmueller.myopenid.com/”>
  <hasAge>32</hasAge>
</rdf:Description>
<rdf:Description
  rdf:about=”http://cmueller.myopenid.com/”>
  <hasArea resource=”&u;Mathematics”/>
</rdf:Description>
<rdf:Description
  rdf:about=”http://cmueller.myopenid.com/”>
  <hasLanguage>English</hasLanguage>
</rdf:Description>
 ...
</rdf:RDF>
```

Figure 6: A User Modeling Ontology and Instantiation in OWL

In Figure 6, we provide a simplified extract of a user modeling ontology in OWL (to the left) as well as its instantiation (to the right), i.e. the representation of user-specific reusable characteristics that can be shared with other systems. Note that the ontology is not conformant with the Gumo specification: For simplicity, we neglect the `range` attribute as well as simplify the Gumo user dimensions. The ontology represents general, i.e., *domain-independent*, user characteristics and has to be extended with mathematical preferences and contexts. Consequently, we embed the general characteristics into our mathematical user model. For illustration purpose, we represent this user model in XML+RDFa (W3C RDFa Primer) as we believe it to be more readable. A user would of course not configure the adaptation system at this level, but via suitable graphical user interfaces or harvesting tools.

```
<profile about="http://cmueller.myopenid.com/">
 <profile type="static">
  <span property="hasAge">32</span>
  <span property="hasLanguage">German</span>
  <span property="hasLanguage">English</span>
  <span property="hasLanguage">French</span>
 </profile>
 <profile type="default">
  <context>
   <span rel="hasArea" href="#Mathematics"/>
   <span property="hasArea"
         href="#Model_Theory"/>
   <span property="hasLanguage">German</span>
  </context>
  <tag xref="ntn123#rend456" weight="5"/>
  <tag xref="ntn244#rend789" weight="1"/>
  ...
  <tag xref="ntn244#rend645" weight="7"/>
</profile>
 ...
 <profile type="situational">
  <context>
   <span rel="hasEvent" href="#GenCS_lecture" />
   <span rel="hasArea" href="#Computer_Science" />
   <span rel="hasAudience" href="#GenCS_students" />
   <span rel="hasLocation" href="#Bremen" />
   <span property="language">German</span>
  </context>
  <tag xref="ntn123#rend889" weight="5"/>
  ...
  <tag xref="ntn244#rend645" weight="7"/>
 </profile>
 <profile type="situational">
  <context>
   <span rel="hasEvent" href="#WSKS_conference" />
   <span rel="hasAudience" href="#WSKS community" />
   <span rel="hasArea" href="#Psychology" />
   <span rel="hasArea" href="#Information_science" />
   <span rel="hasLocation" href="#Athens" />
   <span property="hasLanguage">French</span>
  <context>
 </profile>
</profile>
```

Figure 7: XML+RDFa Representation of a User Model

Figure 7 provides an example of a user model representation in XML+RDFa (W3C RDFa Primer). For convenience, we leave out the namespaces and `type` attribute of the `tag` element. The user model includes a static profile, a default notation model as well as two situational models for a *computer science lecture* and *conference talk*. Each model provides context-parameters describing the user in the specific situation or default scenario: The user is a native German speaker focusing on mathematics, in particular, model theory. In her default setting she prefers German notations, but wants to adapt these for her lecture on computer science. While at a conference in Athens, she wants to switch to a more international scenario that addresses a more global audience from the area of information systems & psychology. As she expects a lot of French researchers to attend the conference, she would like to adapt to their notation systems (thus selecting `French` as language) without having to select the respective notations. Consequently, the situational model for the conference does not include any explicit notation references but leaves it to the conversion algorithm to pick the appropriate ones. In contrast, the two other models (for her lecture and private study) include explicit notation references allowing the user to keep control over the selection of notations.

To represent the user model in XML, we introduce the elements `profile`, `context`, and `tag`. The `profile` element represents different models (specified by a `type` attribute), i.e., the `static` profile, the `default` model, and the `situational` model. The `context` element represents the notation context for both, default and situational models, and comprises RDFa annotations describing the respective context.

5.2 System-Independent Adaptation Support

Our adaptation framework in Section 4.1 is based on the two system-independent components JOMDoc and JOBAD. Outsourcing adaptation functionality into these two central components is our first step towards a generic user modeling framework (see Section 6).

JOMDoc is an open-source Java library for OMDoc developed at the Jacobs University Bremen JOMDoc and provides OMDoc-specific data structures, our notation model and implements the conversion algorithm. The library has been integrated into diverse systems, such as the semantic Wiki SWiM (Lange, 2008) (see Section 6) and the document reader *panta rhei* (panta-rhei) (see Section 5), allowing them to reuse the adaptation functionality.

Apart from the notation adaptation, JOMDoc provides the conversion of OMDoc

```
<omdoc xmlns="http://omdoc.org/ns"
       xmlns:m="http://www.w3.org/1998/Math/MathML"
       xmlns:om="http://www.openmath.org/OpenMath">
 <notation name="minus" xml:id="ntn123">
  <rendering xml:id="rend456"> ... </rendering>
 </notation>
 <notation name="plus" xml:id="ntn123">
  <rendering xml:id="rend789"> ... </rendering>
 </notation>
 <theory>
  <import from="http://openmath.org/cd/arith1#plus" />
  <import from="http://openmath.org/cd/arith1#minus" />
  <m:semantics>
   <m:math>
    <m:mrow>
     <m:mi>a</mi>
     <m:mo>+</mo>
     <m:mi>b</mi>
     <m:mo>−</mo>
     <m:mi>b</mi>
    </m:mrow>
   </m:math>
   <m:annotation−xml>
    <om:OMOBJ xml:id="obj333">
     <om:OMA>
      <om:OMS name="plus" cd="arith1" />
      <om:OMV name="a" />
      <om:OMA>
       <om:OMS name="minus" cd="arith1" />
       <om:OMV name="b" />
       <om:OMV name="c" />
      </om:OMA>
     </om:OMA>
    </om:OMOBJ>
   </m:annotation−xml>
 </theory>
</omdoc>
```

Figure 8: Example content in OMDoc

content into XHTML. During this conversion, JOMDoc can add RDFa annotations to embed the semantics of the rich OMDoc format in the XHTML output. We make use of this feature to preserve the tracked notations definitions that have been applied to convert the included OpenMath expressions, which allows us to extract the respective notation targets for the user modeling described in Section 3.2. We use `tag` elements to represent the previously mentioned references from a mathematical expression to the required rendering elements. Figure 8 provides an example: To generate the parallel markup (representing $a + b - c$) from the OpenMath expression `obj333`, two notation definitions are required, one for the `plus` and one for the `minus` symbol. From both definitions only one rendering is selected and applied to generate the parallel markup. To preserve the conversion information, JOMDoc adds two tags which reference the applied `rendering` children and point to the OpenMath `object`. Figure 9 provides the respective representation of the tags in OMDoc and their translation into RDFa.

JOBAD is the Javascript API for OMDoc-based Active Documents (JOBAD). The framework can be integrated by any system supporting Javascript allowing them to provide adaptive and interactive views on their content. However, only systems that support the OMDoc model and integrate JOMDoc as well as further enabling technologies can

Tag representation in OMDoc

```
<theory>
 <tag xref=”ntn123#rend456” object=”obj333”/>
 <tag xref=”ntn123#rend789” object=”obj333”/>
 <m:semantics>
   ...
   <om:OMOBJ xml:id=”obj333”>
   ...
 </m:semantics>
</theory>
```

Tag representation in XHTML+RDFa

```
<div about=”obj333” xmlns:o=”http://omdoc.org/ns”>
 <span rel=”o:tag” href=”ntn123#rend456” />
 <span rel=”o:tag” href=”ntn123#rend789” />
 <m:semantics>
  ...
  <om:OMOBJ xml:id=”obj333”>
  ...
 </m:semantics>
</div>
```

Figure 9: Tags representation in OMDoc and XHTML+RDFa

make use of the system's suite of services. JOBAD integrates standalone implementations, such as a demonstrator for flexible elisions (Kohlhase, Lange, and Rabe, 2007a) and the formulae search engine MathWebSearch (MathWebSearch), into a general framework. Further information on the development and releases can be found at (JOBAD).

5.3 Adaptive Notations in E-Learning

panta rhei is an interactive and collaborative reader for active documents (Müller and Kohlhase, 2007; panta-rhei). While users are reading, rating, and discussing their documents, implicit and explicit user modeling techniques are applied to personalize the adaptation of content (see (Müller and Kohlhase, 2008b) for details). Figure 10 displays the two constituents of the system: *panta* (the user interface) and *janta* (the backend service). *panta* implements the discussion, annotation, and tagging facilities and gathers information on the user's notation preferences. By integrating JOBAD, *panta* can provide interactive services. *janta* takes over all content and user data handling.

Authors can draw on the sTeX→OMDoc→XHTML workflow (see (Kohlhase, 2008) for details) to write documents in their preferred LaTeX editor and publish their results in *panta rhei*. During the conversion to XHTML, semantic identifiers and metadata are preserved, which improves the web-accessibility of the imported documents. Markup of narrative structure allows us to adapt the size and navigation of documents (see (Kohlhase, Müller, and Müller, 2007b)), markup of concepts allows semantic search and easy cross-linking, enhancement with action triggers facilitates interactivity, and distinction of content and

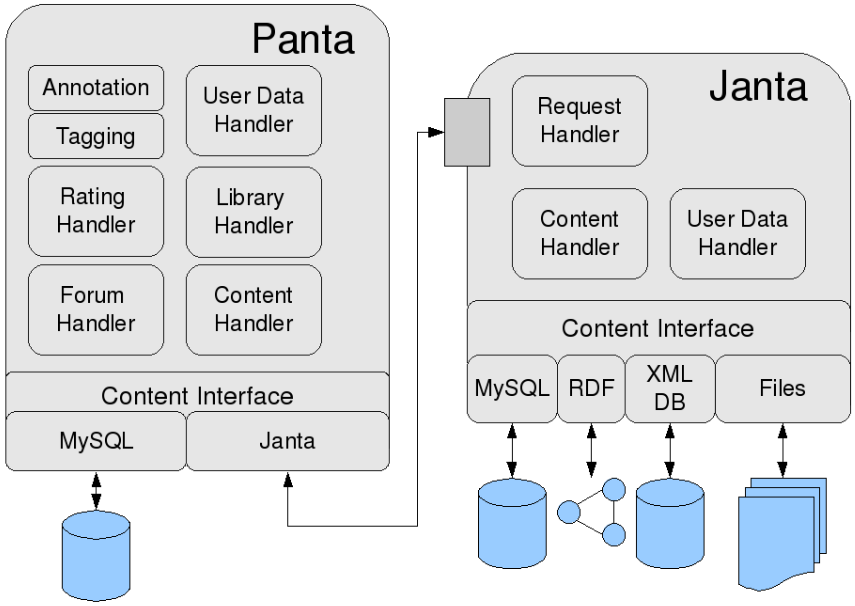

Figure 10: System Architecture

form supports different visualizations. For example, the introductory computer science lecture at Jacobs University Bremen has been written in sTEX and imported to *panta rhei* via JOMDoc. During the import, the lecturer can specify notation preferences, and these are used to generate an initial presentation of the course material, which can then be adapted by the students.

5.4 Evaluation

Informal discussions with mathematical lecturers and researchers has provided valuable feedback for us and revealed further use cases we want to address. Some mathematicians disagreed that notations are context-dependent and vary frequently. They are convinced that mathematical notations are *universal*: They are *standardized within their community* and *alternatives are hardly used and would not be accepted*. However, with their arguments they support our approach, as in the scope of this interview “community” referred to a mathematical sub-community, e.g., the area of *computability*. Eventually, most mathematicians agreed that adapting notations might actually be useful for their students that are not yet well-experienced in the area. However, they doubted that replacing an author’s notation would be useful and rather worried that this would impair the understandability of a text and destroy the author's intention. After all, mathematicians spend much time to select appropriate notations that can be understood and are accepted by the community. However, pointing out difference to a learner model seemed to be useful. Mathematicians agreed that notations are hampering the understanding, but emphasized that this is primarily due to the number of new notations that are introduced in textbooks or papers. Consequently, readers, both students and professionals, have to switch back and forth between a *notation index* and the current text passage. In particular, when studying complicated proofs this really slows down the reading as mental efforts are wasted to memorize new notations. However, some mathematicians believed that a respective reading environment providing natural language terms for the notations, definitions, explanations, or examples (filtered based on the user's experiences and preferably without distracting the user) would increase the efficiency and quality of the reading experience. Please find the "wish list" that, according to the mathematicians, would ease up their life and the learning experience of their students in (Kohlhase et al., 2009).

6 Related Work

6.1 Adaptive Hypermedia & Use Modeling Techniques

In the scope of our work, we have analyzed existing approaches in the area of user modeling and sketch the most influential approaches for our work below.

Generic User Modeling (Kobsa, 2007) provides an overview on generic user model systems that aim at separating user modeling components from other functionality and making them reusable for the development of other user-adaptive systems. Even more forceful, user modeling servers are introduced, which allow to separate and even outsource the user modeling functionality from user-adaptive application systems. For example, the Gumo UserModelService (Heckmann et al., 2005) provides a distributed approach for accessing and storing user information via HTTP requests. Taking on this approach, we aim at an incremental outsourcing of user modeling and adaptation components into centrally available libraries, such as JOMDoc and JOBAD, which will eventually implement our toolkit (Müller and Kohlhase, 2008a).

Context Modeling (Nauerz et al., 2008) has inspired the design of our user models as proposed in Section 3.1, although the authors' approach does not aim at interoperability and exchange of user models. Andreas Nauerz et al. highlight that user models neglect the context users are acting in and can only be regarded suitable models, if the role, interest, and preferences of users do not change over time. The authors propose *context models*, which model the user's preferences for concrete situations. They are manually created by users allowing them to specify *initial settings* as well as *context attributes* (time, date, location) which define when they should become active. While a context model is *active*, all user behaviors, such as tags, are tracked and associated with the model, which is thereby extended implicitly at run time. Users can manually switch between their context models or allow the system to switch automatically to the context model that matches the observed user behaviors best.

User Model Representation (Andrejko et al., 2007) provide an overview on user model representations for web-based information systems. In particular, advantages and disadvantages of *non-ontological* representations, such as sets of relational database tables or XML files (see e.g. the AHA! system (Bra and Calvi, 1998)), and ontological representation formats such as UserML (Heckmann and Krüger, 2003), GUMO (Heckmann et al., 2005), are discussed. The former provide good means for describing user characteristics (XML also facilitates interoperability) but do not offer value from a user modeling perspective. In contrast, ontology representations (based on RDF/ OWL formalism) eliminate disadvantages of XML by defining a vocabulary for defining properties and by supporting automated reasoning.

Adaptive Hypermedia tailors the content of hypermedia systems to users based on their goals, abilities, interests, or knowledge. A plethora of systems exist aiming at guiding users towards relevant information, supporting users to understand the presented information, and changing the presentation to fit a specific platform and environment. During the last decade, adaptive approaches are more and more concerned with enriching their contents via *semantically labelled reusable material*: (Conlan, Lewis, Higel, O'Sullivan, and Wade, 2003) integrate *semantic web services* and adaptive hypermedia. The Adaptive Hypermedia Architecture (AHA) (Bra, Aerts, Berden, Lange, Rousseau, Santic, Smits, and Stash, 2003) models relationships between concepts and expresses prerequisites and suitability of page links. (Cristea, Smits, and de Bra, 2007) integrate the *LOAS framework* and places a great focus on explicit semantics. However, neither approach is suited to represent mathematical knowledge.

6.2 Mathematical (Notation) Modeling

Several attempts have been taken to handle mathematical notation. The two major standards for representing math on the Web, MathML (W3C, 2003) and OpenMath (OpenMath), allow to distinguish content and form of notations and thus to reduce ambiguities and inconsistencies. (Smirnova and Watt, 2006; Naylor and Watt, 2001) have addressed different *notation contexts* that can cause multiple notations of the same mathematical concept, namely *area of application*, *national conventions*, *level of sophistication*, the *mathematical context*, and the *historical period*. They also provided the first approach towards *modeling notation preference*: The author provide a *notation selection tool* (Smirnova and Watt, 2006) that allows users to design *user-specific XSLT stylesheets* for the conversion of mathematical notations based on (W3C, 2003; OpenMath).

Apart from the previously mentioned approaches, only the ActiveMath group (ActiveMath) has started to address mathematical notation representation and modeling. The ActiveMath system (Melis and Siekmann, 2004) is based on our OMDoc format and provides user-adaptivity in the the selection of examples and the sequencing of learning objects into user-specific courses as well as first attempts towards the adaptive presentation of course material. Adaptation is based on user models (Melis, 2001), which represent the competencies of learners regarding specific mathematical concepts. However, notations are only rudimentary covered so far: The ActiveMath system supports lecturer to specify their notation preferences, but do not yet provide a sophisticated adaptation of notations towards single users as notation preferences are not yet part of the learner models. Instead, adaptation is solely based on global metadata such as the degree of abstractness and presentation style (text, formal, graphical). With our work we provide an extension to the ActiveMath model and aim at close cooperation with the ActiveMath group to integrate both approaches, benefiting from the group's experiences in integrating knowledge representation techniques with didactic and psychological modeling approaches as well as in applying learning theory to mathematical education systems. For example, (Melis, Faulhaber, Eichelmann, and Narciss, 2008) proposes a system and domain-independent competency hierarchies, which allow interoperability of learning environments and reuse of learning resources. These hierarchies are used to represent the student's competencies in student models (Faulhaber and Melis, 2008) and allow the pedagogically founded, user-specific selection of course material (see (Ullrich, 2008) for details).

6.3 Notation Case Studies

The semantic Wiki SWiM (Lange, 2008) integrates the central libraries JOMDoc and JOBAD to handle and adapt mathematical notations. In fact, the developer of SWiM, Christoph Lange, is strongly involved in the implementation of both components. SWiM is currently used within two projects: The OpenMath *community* is using the Wiki to refactor their content dictionaries for the upcoming OpenMath3 standard, while the *Flyspeck project* (Hales, McLaughlin, et al., 2007; Lange, McLaughlin, and Rabe, 2008) aims at formalization a mathematical proof (the Kepler conjecture) using this collaborative environment. Both case studies highly depend on respective notation support: The MathML and OpenMath communities want to specify default notations that are commonly accepted by mathematicians as well as alternatives, while the Flyspeck project aims at "crowdsourcing" hundreds of proof sketches to provide a consistent collection which can be incrementally transferred into machine-verifiable and a fully formal representations.

7 Conclusion & Outlook

In this paper, we combine mathematical knowledge management techniques with approaches from the user modeling and adaptive hypermedia. We provide a *novel framework* that allows to adapt mathematical notations based on the user's notation context and preferences. We extended our previous framework (Kohlhase et al., 2008) with user modeling facilities drawing on existing user modeling standards, but extending them to model mathematical practices. Moreover, we introduce an adaptive notation framework building on our existing adaptable presentation algorithm and evaluate it from user feedback.

On the one hand our work contributes to the area of mathematical knowledge management, which provided the foundation to manage and reify mathematical notations, but has largely neglected advanced services such as adaptation so far. On the other hand our work extends existing user modeling approaches by making introducing content/form techniques from MKM that allow notation generation and user adaptability down to the level of individual symbols in formulae. Further work will focus on the integration of the previously proposed features as well as respective implementations and evaluations.

Acknowledgments

We would like to thank the KWARC group for their valuable feedback and discussions. Special thanks go to Stefania Dumbrava, Josip Dzolonga, Christoph Lange, Darko Makreshanski, Dimitar Misev, Normen Müller, Andre Nies, Florian Rabe, Alen Stojanov, and

Jakob Ücker. This work was supported by JEM-Thematic-Network ECP-038208.

References

ActiveMath. ActiveMath, seen September 2008. URL `http://www.activemath.org/`. web page at `http://www.activemath.org/`.

Anton Andrejko, Michal Barla, and Maria Bielikova. Ontology-based user modeling for web-based information systems. In *Advances in Information Systems Development*, pages 457–468. Springer, 2007. URL `http://www.springerlink.com/content/nt7617hp23224774/`.

Dominique Archambault, Bernhard Stöger, Donal Fitzpatrick, and Klaus Miesenberger. Access to scientific content by visually impaired people. *Upgrade*, VIII(2):14 pages, April 2007. ISSN 1684-5285. URL `http://www.upgrade-cepis.org/issues/2007/2/upgrade-vol-VIII-2.html`. Digital journal of CEPIS. A monograph in spanish was published in Novática.

Joachim Baumeister and Martin Atzmüller, editors. *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) Conference Proceedings*, volume 448, 2008.

George Boole. *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities.* Dover Publications NY, Reprinted with corrections, 1858. URL `http://www.gutenberg.org/etext/15114`. Originally published in 1854.

P. De Bra and L. Calvi. AHA: a generic adaptive hypermedia system. In *Proceedings of the 2nd Workshop on Adaptive Hypermedia and Hypermedia*, 1998. URL `http://wwwis.win.tue.nl/ah98/DeBra.html`.

P. De Bra, A. Aerts, B. Berden, B. De Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash. AHA! the adaptive hypermedia architecture. In *Proceedings of ACM Hypertext Conference*, pages 81—84, 2003.

P. Brusilovsky, A. Kobsa, and W. Neidl, editors. *The Adaptive Web: Methods and Strategies of Web Personalization.* Number 4321 in LNCS. Springer Berlin / Heidelberg, 2007. URL `http://www.springerlink.com/content/x646782t122p`.

George J. Chaitin. *Algorithmic Information Theory.* Cambridge University Press, 1987. ISBN 978-0521343060.

Yi-Min Chee, Katrin Franke, Max Froumentin, Sriganesh Madhvanath, Jose-Antonio Magana, Gregory Russell, Giovanni Seni, Christopher Tremblay, Stephen M. Watt, and Larry Yaeger. Ink markup language (inkml). W3C Working Draft, 2006. URL `http://www.w3.org/TR/InkML/`.

O. Conlan, D. Lewis, S. Higel, D. O'Sullivan, and V. Wade. Applying adaptive hypermedia techniques to semantic web service composition. In *Proceedings of International Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 53—62, 2003.

Alexandra Cristea, David Smits, and Paul de Bra. Towards a generic adaptive hypermedia platform: A conversion case study. *Journal of Digital Information*, 8(3), 2007. URL `http://dialnet.unirioja.es/servlet/oaiart?codigo=2415354`.

Hans Cuypers, Arjeh M. Cohen, Jan Willem Knopper, Rikko Verrijzer, and Mark Spanbroek. MathDox, a system for interactive Mathematics. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008*, pages 5177–5182, Vienna, Austria, June 2008. AACE. URL `http://go.editlib.org/p/29092`.

Masahiro Horiand Jerome Euzenat and Peter F. Patel-Schneider. Owl web ontology language: Xml presentation syntax. W3C Note, 2003. URL `http://www.w3.org/TR/owl-xmlsyntax/`.

Arndt Faulhaber and Erica Melis. An efficient student model based on student performance and metadata. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikos Avouris, editors, *Proceedings of the 18th European Conference on Aritifical Intelligence*, pages 276—280, 2008.

Galileo Galilei. *The Assayer*. Rome, October 1623.

Alberto González Palomo. Sentido: an authoring environment for OMDoc. In OMDoc – *An open markup format for mathematical documents [Version 1.2]* Kohlhase (2006), chapter 26.3. URL `http://omdoc.org/omdoc1.2.pdf`.

Thomas C. Hales, Sean McLaughlin, et al. The Flyspeck project. `http://code.google.com/p/flyspeck/`, seen November 2007. URL `http://code.google.com/p/flyspeck/`.

D. Heckmann and A. Krüger. A user modeling markup language (userml) for ubiquitous computing. In P. Brusilovsky, A. Corbett, and F. de Rosis, editors, *Ninth International Conference on User Modeling (UM 2003)*. Springer, 2003.

Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta von Wilamowitz-Moellendorff. Gumo – The General User Model Ontology. In *User Modeling 2005*, pages 428–432, 2005. doi: http://dx.doi.org/10.1007/11527886_58. URL `http://dx.doi.org/10.1007/11527886_58`.

Brent Hendricks and Adan Galvan. The Connexions Markup Language (CNXML). `http://cnx.org/aboutus/technology/cnxml/`, 2007. URL `http://cnx.org/aboutus/technology/cnxml/`. Seen June 2007.

Geneva Henry, Richard G. Baraniuk, and Christopher Kelty. The connexions project: Promoting open sharing of knowledge for education. In *Syllabus, Technology for Higher Education*, 2003. URL `http://www.syllabus.com/summer2003/pdf/T03.pdf`.

JOBAD. JOBAD framework – JavaScript API for OMDoc-based active documents. `https://jomdoc.omdoc.org/wiki/JOBAD`, 2008. URL `https://jomdoc.omdoc.org/wiki/JOBAD`.

JOMDoc. JOMDoc Project — Java Library for OMDoc documents. `http://jomdoc.omdoc.org`. URL `http://jomdoc.omdoc.org`. seen March 2009.

Alfred Kobsa. Generic User Modeling Systems. In Brusilovsky et al. (2007), chapter 4, pages 136–154. URL `http://www.springerlink.com/content/x646782t122p`.

Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006. URL `http://omdoc.org/omdoc1.2.pdf`.

Michael Kohlhase. Using LaTeX as a semantic markup format. *Mathematics in Computer Science*, 2008. URL `https://svn.kwarc.info/repos/stex/doc/mcs08/stex.pdf`.

Michael Kohlhase, Christoph Lange, and Florian Rabe. Presenting mathematical content with flexible elisions. In Olga Caprotti, Michael Kohlhase, and Paul Libbrecht, editors, OpenMath*/ JEM Workshop 2007*, 2007a. URL `http://www.openmath.org/meetings/linz2007/`.

Michael Kohlhase, Christine Müller, and Normen Müller. Documents with flexible notation contexts as interfaces to mathematical knowledge. In Paul Libbrecht, editor, *Mathematical User Interfaces Workshop 2007*, 2007b. URL `http://www.activemath.org/~paul/MathUI07`.

Michael Kohlhase, Christine Müller, and Florian Rabe. Notations for living mathematical documents. In Serge Autexier, John Campbell, J. Rubio, Volker. Sorge, Masakasu Suzuki, and Freek Wiedijk, editors, *Intelligent Computer Mathematics, 9th International Conference, AISC 2008 15th Symposium, Calculemus 2008 7th International Conference, MKM 2008 Birmingham, UK, July 28 - August 1, 2008, Proceedings*, number 5144 in LNAI, pages 504–519. Springer Verlag, 2008. URL `http://omdoc.org/pubs/mkm08-notations.pdf`.

Michael Kohlhase, Christoph Lange, Christine Müller, Normen Müller, and Florian Rabe. Notations for active mathematical documents. KWARC Report 2009-1, Jacobs University Bremen, 2009. URL `http://kwarc.info/publications/papers/KLMMR_NfAD.pdf`.

Christoph Lange. SWiM – a semantic wiki for mathematical knowledge management. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *ESWC*, volume 5021 of *Lecture Notes in Computer Science*, pages 832–837. Springer, 2008.

Christoph Lange, Sean McLaughlin, and Florian Rabe. Flyspeck in a semantic wiki – collaborating on a large scale formalization of the Kepler conjecture, June 2008.

Ming Li and Paul Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 1997. ISBN 978-0387948683.

Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer, January 2007. ISBN 3540378812. URL `http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&path=ASIN/3540378812`.

Mathplayer. Mathplayer ¡display math in your browser¿. web page at `http://www.dessci.com/en/products/mathplayer`. URL `http://www.dessci.com/en/products/mathplayer`.

MathWebSearch. Math Web Search. `http://kwarc.info/projects/mws/`, seen Dec. 2008. URL `http://kwarc.info/projects/mws/`.

E. Melis and J. Siekmann. Activemath: An intelligent tutoring system for mathematics. In *Seventh International Conference 'Artificial Intelligence and Soft Computing' (ICAISC)*, volume 3070 of *LNAI*, pages 91–101. Springer-Verlag, 2004.

E. Melis, A. Faulhaber, A. Eichelmann, and S. Narciss. Interoperable Competencies Characterizing Learning Objects in Mathematics. In *Intelligent Tutoring Systems*, volume 5091 of *LNCS*, pages 416–425. Springer, 2008.

Erica Melis. User model description. DFKI Report, DFKI, 2001.

Bruce Miller. `LaTeXML`: A LaTeX to xml converter. Web Manual at `http://dlmf.nist.gov/LaTeXML/`, seen September2007. URL `http://dlmf.nist.gov/LaTeXML/`.

MKM. International Conference on Mathematic Knowledge Management, seen November 2008. URL `http://www.mkm-ig.org/`. web page at `http://www.mkm-ig.org/`.

Christine Müller. Towards CoPing with Information Overload. In Baumeister and Atzmüller (2008).

Christine Müller and Michael Kohlhase. Towards a community of practice toolkit based on semantically marked up artifacts. In M.D. Lytras et al., editor, *Proceedings of the 1st World Summit of the Knowledge Society: Emerging Technologies and Information Systems for the Knowledge Society*, number 5288 in LNAI, pages 41–50. Springer-Verlag Berlin Heidelberg, 2008a.

Christine Müller and Michael Kohlhase. panta rhei. In Alexander Hinneburg, editor, *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, pages 318–323. Martin-Luther-University Halle-Wittenberg, 2007.

Christine Müller and Michael Kohlhase. Context Aware Adaptation: A Case Study on Mathematical Notations. Research reports, Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland, November 2008b. URL `http://www.cs.auckland.ac.nz/CDMTCS//researchreports/341christine.pdf`.

Andreas Nauerz, Marting Welsch, Fedor Bakalov, and König-Ries. Adaptive Portals: Adapting and Recommending Content and Expertise. In Baumeister and Atzmüller (2008).

Bill Naylor and Stephen M. Watt. Meta-Stylesheets for the Conversion of Mathematical Documents into Multiple Forms. In *Proceedings of the International Workshop on Mathematical Knowledge Management*, 2001. URL http://www.emis.de/proceedings/MKM2001.

A.M. Odlyzko. Tragic loss or good riddance? the impending demise of traditional scholarly journals. *International Journal of Human-Computer Studies*, 42:71–122, 1995.

OMCD-Core. OpenMath content dictionaries. web page at http://www.openmath.org/cd/, seen June2008. URL http://www.openmath.org/cd/.

OpenMath. OpenMath Home. http://www.openmath.org/, seen March 2007. URL http://www.openmath.org/.

panta-rhei. The panta rhei Project. URL http://trac.kwarc.info/panta-rhei. seen March 2009.

RDF. Resource description framework (RDF). http://www.w3.org/RDF/, 2004. URL http://www.w3.org/RDF/.

Ron Rivest, Adie Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. URL http://www.matha.mathematik.uni-dortmund.de/~fv/diplom_i/ars78.ps.

Elena Smirnova and Stephen M. Watt. Notation Selection in Mathematical Computing Environments. In *Proceddings Transgressive Computing 2006: A conference in honor of Jean Della Dora (TC 2006)*, pages 339–355, Granada, Spain, 2006.

Carsten Ullrich. *Pedagogically Founded Courseware Generation for Web-Based Learning*. LNCS. Springer, September 2008. ISBN 978-3-540-88213-8. URL http://www.springerlink.com/content/k604618p5351/.

W3C. Mathematical Markup Language (MathML) Version 2.0 (Second Edition). http://www.w3.org/TR/MathML2/, 2003. URL http://www.w3.org/TR/MathML2/. Seen July 2007.

W3C RDFa Primer. RDFa Primer. http://www.w3.org/TR/xhtml-rdfa-primer/, 2008. URL http://www.w3.org/TR/xhtml-rdfa-primer/.