

Towards Community of Practice Support for Interactive Mathematical Authoring

Marc Wagner¹, Christine Müller²

¹Universität des Saarlandes
D-66123, Saarbrücken, Germany
wagner@ags.uni-sb.de

²Jacobs University Bremen
D-28759, Bremen, Germany
c.mueller@jacobs-university.de

In order to foster the use of proof assistance systems, we integrated the proof assistance system Ω MEGA with the standard scientific text-editor $\text{T}_{\text{E}}\text{X}_{\text{M}}\text{A}^{\text{C}}\text{S}$ using the mediator $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$. We aim at a document-centric approach to formalizing and verifying mathematics and software. Assisted by the proof assistance system, the author writes her document entirely inside the text-editor in a language she is used to, that is a mixture of natural language and formulas in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ style. We developed a basic mechanism that allows the author to define her own notation inside a document in a natural way, and use it to parse the formulas written by the author as well as to render the formulas generated by the proof assistance system. This paper examines how the mediator $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$ can be extended to identify, analyze and support communities of practice based on the notation practice of individual authors.

1 Introduction

Mathematical proof assistance systems have not yet achieved recognition and relevance in mathematical practice. Significant progress is still required, in particular with respect to the usability of these systems. Rather than developing a new user interface for the mathematical assistance system Ω MEGA the $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$ system [pla] presents a generic way of integrating proof assistance systems into scientific text-editors by using a flexible and parametric semantic annotation language. $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$ allows for the incremental development of mathematical documents in professional type-setting quality by propagation of changes and context sensitive service menu interaction.

The aim of the $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$ system is to support the complete authoring process of a mathematical document - from creation through formalization to publication - in a collaborative environment. This paper examines the added values for the authoring process when integrating a management module for communities of practice into $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$. The concrete research questions addressed by this paper are: How can we extract the notation profile of an individual author? How can we identify communities of practice? What kind of added values can we provide to these communities in the process of interactive mathematical authoring?

The paper is organized as follows: Section 2 defines the notions of *notation practice* and *community of practice* in the scope of this paper. In Section 3 we present the $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$ system that mediates between the mathematical assistance system Ω MEGA and the scientific text-editor $\text{T}_{\text{E}}\text{X}_{\text{M}}\text{A}^{\text{C}}\text{S}$. Inspired by notational definitions in text-books, we illustrate the possibilities provided to the author to define her own notation. Section 4 describes how the $\text{P}^{\text{L}}\text{A}^{\text{T}}\Omega$ system can be extended to identify, analyze and support *mathematical communities*. Finally, we discuss the added values for the individual author in the process of interactive mathematical authoring before concluding in Section 5.

2 Mathematical Notations and Communities

In our document-centric philosophy the mathematical document authored by the user is the central source of information for the whole system. By processing the specified notations in a document and all referenced ones, the PLATΩ system is able to *extract* the semantics of the mathematical knowledge contained in the document. Furthermore, the document is automatically adapted in case of notational changes. Hence, the notation context can be considered as a dynamic parameter separated from the content of a mathematical document. This facilitates both the identification of the author's *notation practice* and of *communities of practice* (COP) that share specific notation preferences, and conversely, the adaptation of documents to the notation preferences of identified COPs, e.g. groups of readers and co-authors.

Notation Practice: In the scope of this paper, a *notation practice* is the selection of an adequate presentation for symbols. In this sense, an author's *notation practice* is her individual way of selecting notations, which she acquires throughout her life and which is influenced by a number of factors.

Various researchers have addressed the challenge of identifying and describing mathematical practice. In [SW06] Watt and Smirnova introduce possible *reasons* for multiple notations of the same mathematical concept, namely *area of application*, *national conventions*, *level of sophistication*, the *mathematical context*, and the *historical period*: For example, to introduce an imaginary unit, a mathematician uses the symbol i . In contrast, an electrical engineer uses j to avoid confusion with the symbol I for electric current. i and j are two alternative presentations for the symbol “imaginary unit”. Hence, the *notation context* of a mathematical document will most likely refer to the notation i , while the notation selected for a document by an electrical engineer refers to j . Based on the *national convention* we distinguish notations that are commonly used in different languages. For example, a German researcher presents the symbol “binomial coefficient” with the notation $\binom{n}{k}$. In contrast, a Russian researcher uses the alternative presentation C_k^n , while a French researcher will most likely use C_n^k .

In contrast to the five “*reasons*” in [SW06], the OMDOC-based eLearning environment ACTIVEMATH [MLUM05] distinguishes four *categories* that influence the adaptation of notations, namely *language*, *different patterns of the argument*, the *author's style*, and *notations of the same collection*. For example, depending on her individual style, an author will use a/b or $a : b$ (cf. [MLUM05]). In addition, the ACTIVEMATH group specified a prioritization of notation collections, which we referred to as *notation contexts*: *system defaults*, *author*, *book*, *group*, and *individual style* (highest priority). Furthermore, in ACTIVEMATH authors can annotate different layout preferences for notations, such as *color*, *font*, or *border*, which shall also be considered parts of the *notation practice*.

Communities of Practice: In [KK06] Andrea Kohlhasse and Michael Kohlhasse propose the application of the economic theory of *communities of practice* [Wen05] to the area of mathematics. According to their discussions, *mathematical practice* is *inscribed* into documents, e. g. by selecting specific notations or referencing other mathematical publications. Analyzing a *collection of documents* will potentially lead to *clusters of shared practices*, i.e. *communities of practice*, that define the mathematical practice of researchers. For example, the users of a particular proof assistance system establish a *community of notation practice* by adapting to the notation of other users and, vice versa, by influencing the community with their individual notation habits. The longer a user interacts with a certain group of users, i.e. by reading their documents or collaboratively writing papers, the more likely she will adapt to this set of shared notations, hence moving from the outer border of the COP to the inner parts. However, [KK06] further emphasizes the problem of *inscribing practices*, e. g. by *enforcing* the use of specific notations in a system, and calls for the implementation of workflows that preserve the *fluid movement* of users in and between COPs.

This paper aims at resuming the discussion on mathematical COPs by proposing a module for the PLATΩ system which preserves the *dynamics* and *flexibility* of *mathematical communities* in terms of their *notation practice*. By observing and analyzing the communication inside the mediator, the module should be able to dynamically identify and analyze the *notation practice* of individual authors. Once having identified COPs, the module can actively support the community members e.g. by suggesting standard notation, by notifying about conflicts or even by translating documents between COPs.

3 The Mediator PLATΩ

One standard activity in mathematics and areas that are based on mathematics is the preparation of documents using some standard text preparation system like \LaTeX . $\text{\TeX}_{\text{MACS}}$ [vdH01] is a scientific text-editor in the WYSIWYG paradigm that provides professional type-setting and supports authoring with powerful macro definition facilities like those in \LaTeX . As a first step towards assisting the authoring of mathematical documents, we integrated the proof assistance system ΩMEGA into $\text{\TeX}_{\text{MACS}}$ using the generic mediator PLATΩ [WAB06]. In this setting the formal content of a document must be amenable to machine processing, without imposing any restrictions on how the document is structured, on the language used in the document, or on the way the document can be changed. The PLATΩ system [Wag06] transforms the representation of the formal content of a document into the representation used in a proof assistance system and maintains the consistency between both representations throughout the changes made on either side. Fig. 1 shows the architecture of the system integration.



Figure 1: Architecture of the integration of $\text{\TeX}_{\text{MACS}}$ and the ΩMEGA system via PLATΩ

Such an integrated authoring environment should allow the user to write her mathematical documents in the language she is used to, that is a mixture of natural language and formulas in \LaTeX style with her own notation. To understand the meaning of the natural language parts in a mathematical document we currently rely on annotations for the document structure that must be provided manually by the user. Although it might still be acceptable for an author to indicate the macro-structures like theories, definitions and theorems, writing annotated formulas (e.g. “ $\text{\F{in}}\{\text{\V{x}}, \text{\F{cup}}\{\text{\V{A}}, \text{\V{B}}\}\}$ ” instead of “ $x \in A \cup B$ ”) is definitely not.

Aiming at a document-centric approach to formalizing mathematics, we presented a *dynamic notation* mechanism [AFNW07] that allows authors to define their own notation and to use it when writing formulas within the same document. Furthermore, this mechanism enables the proof assistance system to access the formal content and use the same notation when presenting formulas to the author. Thus, the notation context \mathcal{N} is on the one hand the key to extract the semantic content of a document \mathcal{D} , on the other hand by changing the notation context the document is automatically adapted to the desired notation by partially re-rendering the semantic content. Hence, in the context of interactive mathematical authoring the notation context of a document is important for both: the system to *understand* a document and the author to *translate* a document into her own context.

Notations defined by authors are typically not specified as grammar rules. Therefore, we introduced a user friendly WYSIWYG method to define notations and to automatically generate grammar rules from it. Looking at standard mathematical textbooks, one observes sentences like “*Let x be an element and A be a set, then we write $x \in A$, x is element of A , x is in A or A contains x .*”. Supporting this format requires the ability to locally introduce the variables x and A in order to generate grammar rules from a notation pattern like $x \in A$. Without using a linguistic database, patterns like x is in A are at the moment only supported as pseudo natural language.

We introduced the following annotation format to define the operator \in and to introduce multiple alternative notations for \in as closely as possible to the textbook style.

```

\begin{definition}{Predicate  $\in$ }
  The predicate  $\text{\concept{\in}{elem \times set \rightarrow bool}}$ 
  takes an individual and a set and tells whether that
  individual belongs to this set.
\end{definition}

\begin{notation}{Predicate  $\in$ }
  Let  $\text{\declare{x}}$  be an individual and  $\text{\declare{A}}$  a set,
  then we write  $\text{\denote{x \in A}}$ ,  $\text{\denote{x is element of A}}$ ,
   $\text{\denote{x is in A}}$  or  $\text{\denote{A contains x}}$ .
\end{notation}

```

4 Towards COP Support in PLAT Ω

We propose a module for the PLAT Ω system which should identify and analyze the *notation practice* of individual authors in order to support *mathematical communities* in terms of their *notation practice*. To associate the communication going through the PLAT Ω system to particular users, it is necessary to extend the current session management by a user and community management. Thereby we have to distinguish between *explicit* COPs, which are actively created by users, and *implicit* COPs, which are continually computed by observing and analyzing the communication. In the following we will outline the target features of the proposed COP support module that will be built on top of the *dynamic notation* mechanism of PLAT Ω .

4.1 Explicit CoPs

Creating static CoPs. The COP module allows users to create a new COP in the system by entering a name, e.g. a working group, and a list of members, e.g. the work colleagues. In addition, users extensionally describe the new COP by selecting the COP specific library, i.e. the list of documents with notations commonly used and accepted by the working group.

Maintaining static CoPs. Although we use the notation of a static COP to distinguish them from the automatically computed clusters and practices, COPs are always emergent structure and change over time. For example, authors may introduce new notations in their new documents, which triggers the automatic extension of a COP's notation. To support the dynamics of static COPs, PLAT Ω allows users to manually maintain and adapt the notations over time, i.e. to add or delete notations from the COP library and, in particular, to ensure an up-to-date COP library. By observing and analyzing the members notation choices, the system can optionally recommend to add or delete notations if they are shared respectively no longer used by most of the members, or it may even automatically adapt the COP library.

Creating dynamic CoPs. Given a fixed set of COP members and a name of the COP, the PLAT Ω module may also identify the COPs notations dynamically: a member's notation choices are captured and compared to the ones made by his colleagues while they are using the PLAT Ω system. This results in an automatically identified set of *shared notation practices* that describe the new COP library, which is maintained and continuously adapted by the new COP module.

Maintaining dynamic CoPs. Users may evaluate the computed set of notation practices: Shared notation practices that are not relevant for the COP can be deleted from the COP's library. In addition, notations may be added although they are not yet commonly shared among the members.

CoP memberships. Users are most likely members of multiple CoPs and may acquire memberships via three optional ways: (1) They can be added to a CoP by other users, e.g. while creating a CoP. (2) They can actively join existing CoPs and terminate their membership. (3) Finally, by matching a user’s notation practice to the ones that describe particular CoPs, the CoP support module may recommend users to join CoPs.

Support. When opening a document \mathcal{D}_i which has been written in the notation \mathcal{N}_i , authors have to indicate in which CoP context the document shall be processed, i.e. which notation \mathcal{N}_j shall be used to display and edit the document. If there is no CoP specific notation \mathcal{N}_j available, then the original notation \mathcal{N}_i of the document is used by default. This *translation* procedure is applied to the transitive closure of all referenced documents. The mechanism ensures the notation consistency during the collaborative editing of a document by authors which are members of the same CoP. Conversely, authors which are members of different CoPs are allowed to collaboratively edit the same document while using different notation contexts. Indeed, changes to the document are transmitted to all authors after being transformed into the specific notation context of the respective author.

Example Scenario. Based on the *national convention* we distinguish notations that are commonly used in different languages. For example, a German researcher presents the symbol “binomial coefficient” with the notation $\binom{n}{k}$. In contrast, a Russian researcher uses the alternative presentation C_k^n , while a French researcher will most likely use C_n^k . Thus, we have three different authors which are members of three different national CoPs. If the German researcher for example opens a document written by the Russian researcher, the document is automatically translated into the German notational context, in particular all instances of C_k^n are converted into instances of $\binom{n}{k}$. Furthermore, when the German researcher, for example, modifies this document and sends it back to the Russian researcher, all modifications are translated into the Russian notational context. Hence, the proposed module improves the collaborative authoring across national borders.

4.2 Implicit CoPs

In contrast to the creation of *explicit* CoPs, the proposed module may also automatically identify *implicit* CoPs and their notations without setting a fixed group of users. Similarly to the automatic identification of notations of explicitly created CoPs, the notation choices of all users are monitored and matched to find clusters of similar practices. These notation clusters may be interpreted as *implicit* CoPs. However, automatically identified CoPs are only handled and maintained internally.

Identification. The *implicit* CoP of an individual author (a_0) is computed on the fly by comparing the documents (\mathcal{D}_i) and notational contexts (\mathcal{N}_j) opened or modified by the author with the ones opened or modified (*known*) by other authors. This results in a list of authors ordered by the percentage of match that we truncate below 50%.

$$\text{dCoP}(a_0) = \{ \langle p, A \rangle \mid p \geq 50\% \wedge \forall a_i \in A. p = \frac{\#(\mathcal{D}_k, \mathcal{N}_j) \text{ known by } a_0 \text{ and } a_i}{\#(\mathcal{D}_k, \mathcal{N}_j) \text{ known by } a_0} \}$$

Support. If an author opens a document for which there does not exist a notation context in an *explicit* CoP the author holds a membership of, the *implicit* CoP of the author is scanned to find the author with the highest percentage of match who also has opened that document. The document is then translated into the notation context used by that author. If there is no such author, the document is opened with its default notation context. This procedure is transitively applied to each referenced document. In the case of collaborative editing, authors that are not members of any *explicit* CoP are by default treated as members of individual CoPs. Additionally, they are notified about conflicts in their notation context.

Example Scenario. Let \mathcal{D}_i be a document and \mathcal{N}_j be a notation context, then consider the situation:

Author A	Author B	Author C	Author D
$(\mathcal{D}_1, \mathcal{N}_1)$	$(\mathcal{D}_1, \mathcal{N}_2)$	$(\mathcal{D}_1, \mathcal{N}_2)$	$(\mathcal{D}_1, \mathcal{N}_2)$
$(\mathcal{D}_2, \mathcal{N}_3)$	$(\mathcal{D}_2, \mathcal{N}_4)$		$(\mathcal{D}_2, \mathcal{N}_4)$
		$(\mathcal{D}_3, \mathcal{N}_5)$	$(\mathcal{D}_3, \mathcal{N}_5)$

The dynamic COPs in this situation are computed as follows:

Author A	\emptyset
Author B	$\{ < 100\%, \{ Author D \} >, < 50\%, \{ Author C \} > \}$
Author C	$\{ < 100\%, \{ Author D \} >, < 50\%, \{ Author B \} > \}$
Author D	$\{ < 66\%, \{ Author B, Author C \} > \}$

If for example Author A opens the document \mathcal{D}_3 written by Author C and the document \mathcal{D}_3 refers to the document \mathcal{D}_1 , then \mathcal{D}_3 is partially translated using the notation context \mathcal{N}_1 instead of \mathcal{N}_2 for the parts concerning the referenced document \mathcal{D}_1 . The remaining parts of the document are opened with the default notation context of Author C. Conversely, if Author C opens the document \mathcal{D}_2 , then this document is translated using the notation context \mathcal{N}_4 of Author D because this author has the highest percentage of match with Author C and furthermore Author D already knows that document. Hence, whenever no *explicit* COP information is available, the *implicit* COP is computed for an individual author in order to translate unknown documents into the notation context which is most likely preferred by that author.

5 Conclusion and Future Work

We have examined the integration of a COP support module into the elaborated interactive mathematical authoring framework PLATΩ. The envisaged features including automatic translation between different COPs as well as consistency checks for COPs have been discussed in this paper. The next step is the specification and implementation of the herein described module. By accomplishing this task, the authors are confident in both identifying further aspects of COPs and continuing improving the support for COPs in the process of interactive mathematical authoring.

6 References

- [AFNW07] Serge Autexier, Armin Fiedler, Thomas Neumann, and Marc Wagner. Supporting user-defined notations when integrating scientific text-editors with proof assistance systems. In Manuel Kauers, Manfred Kerber, Robert Miner, and Wolfgang Windsteiger, editors, *Towards Mechanized Mathematical Assistants*, LNAI. Springer, June 2007.
- [KK06] Andrea Kohlhasse and Michael Kohlhasse. Communities of Practice in MKM: An Extensional Model. In Jon Borwein and William M. Farmer, editors, *Mathematical Knowledge Management, MKM'06*, number 4108 in LNAI, pages 179–193. Springer Verlag, 2006.
- [MLUM05] Shahid Manzoor, Paul Libbrecht, Carsten Ullrich, and Erica Melis. Authoring Presentation for OPENMATH. In Michael Kohlhasse, editor, *Mathematical Knowledge Management, MKM'05*, number 3863 in LNAI, pages 33–48. Springer Verlag, 2005.
- [pla] Interactive mathematical authoring with PLATΩ. System homepage at <http://www.ags.uni-sb.de/plato>.
- [SW06] Elena Smirnova and Stephen M. Watt. Notation Selection in Mathematical Computing Environments. In *Proc. Transgressive Computing 2006: A conference in honor of Jean Della Dora (TC 2006)*, Granada, Spain, 2006.
- [vdH01] Joris van der Hoeven. Gnu T_EX_{MACS}: A free, structured, wysiwyg and technical text editor. Number 39-40 in Cahiers GUTenberg, May 2001.
- [WAB06] Marc Wagner, Serge Autexier, and Christoph Benzmlüller. PLATΩ: A mediator between text-editors and proof assistance systems. In C. Benzmlüller S. Autexier, editor, *7th Workshop on User Interfaces for Theorem Provers (UITP'06)*, ENTCS. Elsevier, August 2006.
- [Wag06] Marc Wagner. Mediation between text-editors and proof assistance systems. Diploma thesis, Saarland University, Saarbrücken, Germany, 2006.
- [Wen05] Etienne Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 2005.