Using TeXmacs in Math Education: An exploratory Study

Marc Wagner and Henri Lesourd
Universität des Saarlandes
D-66123, Saarbrücken, Germany
{wagner|henri}@ags.uni-sb.de

In order to foster the use of the standard scientific text-editor GNU TeXmacs in math education and to develop semantic-based added values during the authoring process, we conducted an exploratory study with a first year mathematics course. We developed an extension of TeXmacs to support creating, browsing, solving and submitting exercises to an external repository. During a supervised classroom exercise we collected successive snapshots of the student solutions. We present an analysis of this corpus with respect to the edit behaviour and selected linguistic characteristics. Furthermore, we discuss how the collected information can be used to provide support services for the overall authoring process. The progress of the study is published regularly at www.imath.eu.

1 Introduction

The vision of a powerful mathematical assistance environment, providing computer-based support for most tasks of a mathematician, has stimulated new projects and international research networks in recent years across disciplinary boundaries. A mathematical assistance system that really supports mathematicians in their daily work has to be highly user oriented. The mathematician is used to formulate his problems with pen and paper, we do not want to disrupt this habit, thus we decided to integrate our conception of an ideal mathematical assistance system (IMAS) in the traditional workflow, namely sketching the solution using pen and paper and afterwards encoding the solution and refining the details by means of our TeXmacs [11] based interactive user interface.

Our aim is to build document-centric services that support the author in the text-editor TeXmacs while preparing a document in a publishable format. We want to start with mathematical documents written without any restrictions and try to extract the semantic content with natural language analysis techniques and accordingly generate or modify parts of the document using natural language generation. To analyse the requirements for the natural language techniques we first have to collect in an exploratory study a *corpus*, that is a collection of sample documents.

Regarding the capabilities of state of the art proof assistance systems like ISABELLE [9], COQ [4] or ΩMEGA [10], it seems more realistic to provide the authoring assistance to a first year mathematics student rather than to a mathematician working on the edge of research. Therefore we decided to conduct an exploratory study with a first year mathematics course.

The paper is organized as follows: Section 2 introduces TeXmacs as extendable WYSIWYG authoring environment. Section 3 presents our general architecture consisting of TeXmacs, an exercise management plugin and an external repository. The setting for the exploratory study is described in Section 4. Section 5 reports on a corpus analysis with respect to the edit behaviour and selected linguistic characteristics. We discuss potential added values in Section 6 and summarize the results in Section 7.

2 TeXmacs in a Nutshell

The scientific WYSIWYG (what-you-see-is-what-you-get) text-editor TeXmacs offers scientists a free and user friendly authoring environment. TeXmacs is optimized for efficient input of mathematical formulas and for structuring mathematical documents. High-quality typesetting algorithms and TeX fonts produce professional documents. Inspired by the authoring system TeX [7] and its extension LaTeX [8], TeXmacs produces documents which match the standards of professional publications. Indeed, printed TeXmacs documents are not distinguishable from those produced with LaTeX.

Furthermore, TeXmacs is suitable as an interface for computer algebra systems, as the high typesetting quality goes through for automatically generated formulas. It can be highly customized as it supports the Guile/Scheme extension language. TeXmacs supports the export of documents to PostScript [1] and PDF [2] and offers both import and export to HTML, LaTeX and XML. The system runs on all major Unix platforms and Microsoft Windows.

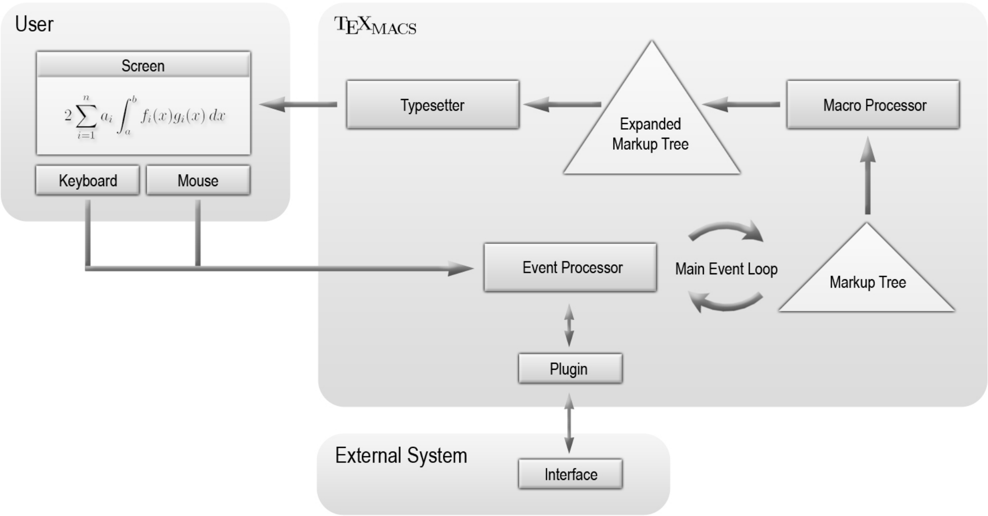


Figure 1: Architecture of the TeXmacs system

We interact with the user through menu-style documents. Therefore we discuss now the architecture of the TeXmacs system (see Figure 1) that allows us to programmatically manipulate documents. For our purposes, the two most important components of the architecture are the macro processor and the event processor. The *macro processor* expands the user-defined macro markup tags in a document tree and translates them into primitive, macro-free TeXmacs markup. One can define new macro tags and the corresponding TeXmacs macros to determine the *physical semblance* of these new tags. Once expanded, the rewritten tree is displayed in the current output device by the *typesetter*. A TeXmacs tree can be displayed in different ways, such as in the markup language, in LaTeX syntax or as a Scheme s-expression.

The *event processor* detects the user input events and updates the current markup tree accordingly. Once updated, the current tree is then expanded and again typeset. One can define new *user input handlers* and write a TeXmacs *plugin* to define the corresponding new *actions* that are to be performed on the current markup tree in the editor. Such plugins are extensions to the event processor, and can be written using the Scheme scripting language, which is embedded in TeXmacs. They can also be employed to allow external systems such as computer algebra systems or proof assistance systems to modify the current markup tree.

The scientific text-editor TeXmacs provides with its graphical WYSIWYG user interface a simple and intuitive access to the power of LaTeX. For a detailed introduction to the LaTeX features of TeXmacs we would like to refer to the tutorials specifically written for the participants of our study. These TeXmacs primers are available at `www.imath.eu`.

3 Exercise Management

The general idea is to use the text-editor as a user interface (UI) for creating, browsing, solving and submitting exercises. The main requirement for an exercise manager is the storage of successive versions of exercise sheets, exercises and their student-specific solutions. Therefore we decided to use the client-server architecture shown in Figure 2, where the internals of TeXmacs, as explained in section 2, are represented by one combined module called *Layout Engine*. The job of TeXmacs is to display the user interface (UI) for navigating through exercises and solutions. Using the advanced layout capabilities of TeXmacs we implemented the interface as a plugin that dynamically manipulates documents. Details about that technique are explained in Subsection 3.1. The iMath server acts as a simple transactional versioning system with basic rights management. User accounts and document versioning information are maintained in a database by the server while the document snapshots are physically stored in the file system. The provided interface is described in Subsection 3.2.

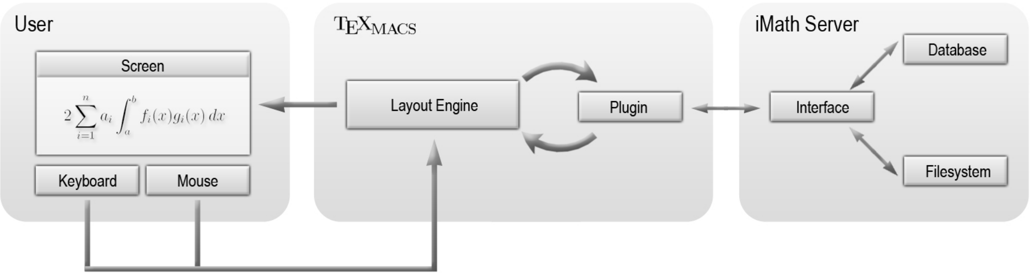
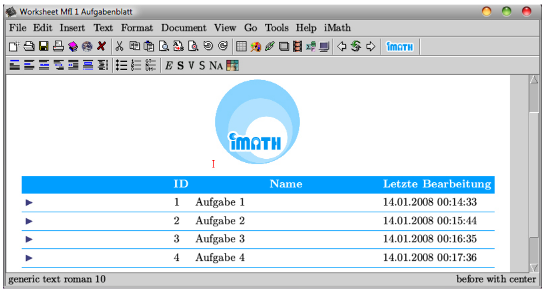


Figure 2: Architecture of the Exercise Manager

3.1 Dynamic Documents as Interfaces

The question was how to navigate inside TeXmacs through the exercises and their solutions. Our answer can be seen as the equivalent of AJAX [6]-driven browser interfaces for modern text-editors. In TeXmacs we encode the menu of the exercise manager as table in a document, using hyperlinked objects as buttons that trigger plugin interface functions on the server. The sophisticated layout engine is able to display this kind of menu in arbitrary styles (see Figure 3). Most importantly we use the DOM-like API of TeXmacs for dynamically updating the document and thus the menu when the user interacts with the interface. Status information can easily be synchronized between server and interface using a cronjob in TeXmacs.


Worksheet Mfl 1 Aufgabenblatt
File Edit Insert Text Format Document View Go Tools Help iMath
iMATH
generic text roman 10
before with center

	ID	Name	Letzte Bearbeitung
▶	1	Aufgabe 1	14.01.2008 00:14:33
▶	2	Aufgabe 2	14.01.2008 00:15:44
▶	3	Aufgabe 3	14.01.2008 00:16:35
▶	4	Aufgabe 4	14.01.2008 00:17:36

Figure 3: TeXmacs UI of the Exercise Manager

3.2 Automatic Versioning of Exercises and Solutions

Since we are interested in successive snapshots of the documents written in the text-editor, we use a server-based persistent storage for exercise sheets, exercises and their solutions. User accounts and document versioning information are stored in a MySQL database while the document content is stored in the file system. Admin users can browse, create, delete and rename exercise sheets and exercises. Additionally, they can watch the user solutions via a replay mechanism, simulated by successive checkouts of subsequent solution versions. Standard users can browse exercise sheets and exercises, and create solutions. A cronjob automatically commits new versions of the solution to the iMath server. Thus the last version is automatically submitted and we are able to monitor the edit behaviour over time.

4 Exploratory Study

For our study we had the great opportunity to collaborate with the first year mathematics course *Mathematik für Informatiker* of Prof. John at Saarland University. In the following we will explain the details and workflow of the study and give an analysis of the authoring behaviour over time.

4.1 Parameters and Workflow

In the first week we introduced TeXmacs to the students who all were completely unfamiliar with LaTeX. The students got quickly familiar with the intuitive usage provided by the WYSIWYG interface of TeXmacs which was a key requirement for being able to conduct this study at all. In the following ten weeks the students had to solve and submit one out of four exercises per week with TeXmacs. After that training period we conducted the real study, a supervised classroom exercise that had to be completely solved with TeXmacs. 49 students had 90 minutes time to solve the following 4 exercises:

1. Let $(a_n)_{n\in\mathbb{N}}$ be a convergent sequence in $\mathbb{R}$ with limit $a \in \mathbb{R}$. Show that it is a Cauchy sequence.

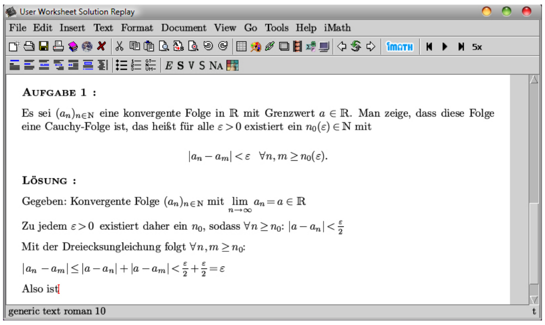
2. Compute the following limits of complex-valued sequences: $\lim_{n\to\infty} \frac{2in^3-n^4}{n^4+3in^2-1}$, $\lim_{n\to\infty} \frac{3n^2-2}{n^3+1}$, $\lim_{n\to\infty} \frac{5n-7n^2}{(n+1)^2-8n}$.

3. For $x \in \mathbb{R}_+$ compute $lim_{n\to\infty}\frac{x^n-n}{x^n+n}$.

4. Compute the following limits of the sequences $(x_n)_{n\in\mathbb{N}}$ with the help of limits proven in the lecture: $x_n = (1-\frac{1}{n})^n$, $x_n = (1-\frac{1}{n^2})^n$.

Our software took automatic snapshots of the current state of the solutions every 10 seconds. Please note that we had no influence on the type of exercises at all, the exercise sheet was completely regular. The concrete exercise sheet as well as more details of the study are available at `www.imath.eu`.

4.2 Analysis of the Authoring Behaviour


User Worksheet Solution Replay
File Edit Insert Text Format Document View Go Tools Help iMath
iMATH
5x
E S V S NA
generic text roman 10
t

Aufgabe 1 :

Es sei $(a_n)_{n\in\mathbb{N}}$ eine konvergente Folge in $\mathbb{R}$ mit Grenzwert $a \in \mathbb{R}$. Man zeige, dass diese Folge eine Cauchy-Folge ist, das heißt für alle $\varepsilon > 0$ existiert ein $n_0(\varepsilon) \in \mathbb{N}$ mit

$$|a_n - a_m| < \varepsilon \quad \forall n, m \geq n_0(\varepsilon).$$

Lösung :

Gegeben: Konvergente Folge $(a_n)_{n\in\mathrm{N}}$ mit $\lim_{n\to\infty} a_n = a \in \mathbb{R}$

Zu jedem $\varepsilon > 0$ existiert daher ein n_0, sodass $\forall n \geq n_0$: $|a - a_n| < \frac{\varepsilon}{2}$

Mit der Dreiecksungleichung folgt $\forall n, m \geq n_0$:

$|a_n - a_m| \leq |a - a_n| + |a - a_m| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$

Also ist

Figure 4: Exercise Replay in TeXmacs

By using the replay mechanism (see Figure 4) specifically developed for this study we were able to analyse the authoring behaviour of the students per exercise over time. First we observed that the students spend on average 30% of their time with searching relevant definitions and theorems in the course material, another 30% with producing a draft solution on paper, and the final 40% with authoring the solution in TeXmacs. We believe that it is important to acknowledge the valuable part of the good old pen & paper method in order to build a system that is really accepted by students as well as working mathematicians. Analysing the authoring behaviour inside TeXmacs, we were able to identify the following types of modifications:

- The standard type of modification is the *monotonic increasing* of the solution when the student is continuously writing up.

- Quite often *Copy & Paste* is used to shorten the complicated input of long mathematical formulas that differ only slightly from previous formulas.

- *Local modifications* can be observed regularly. This means that parts of the last sentence are deleted, rewritten or corrected, like for example adding missing brackets in formulas.

- *Global modifications* can only be observed in very few cases. Refactoring operations like variable renaming affect in general more than one sentence. Sometimes such an operation is not executed but written declaratively, for example ”*By renaming ε_2 to ε the theorem follows.*”.

5 Corpus Analysis

The documents collected during the experimental study have been analysed with respect to different linguistic aspects. In the following we will report in more detail about the analysis of the most important aspects *formula verbalisation*, *style of sentences*, *concluding step* and *justifications*.

5.1 Formula Verbalisation

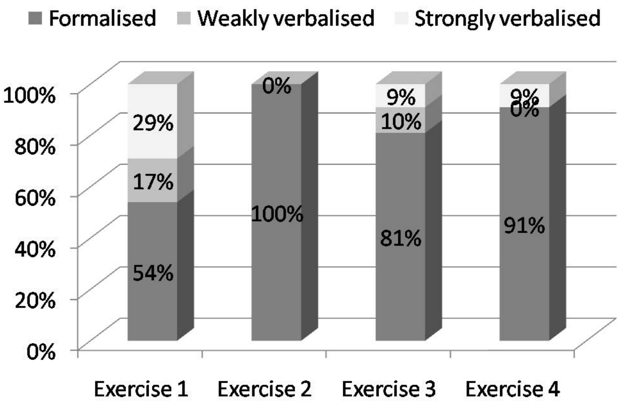


Figure 5: Formula Verbalisation

The aspect formula verbalisation is divided into the following categories: (1) *formalised* (e.g. “$lim_{n\rightarrow\infty}a_n = a$”, “$|a_n - a_m| < \varepsilon \quad \forall n,m \geq n_0(\varepsilon)$”) where formulas are completely written in symbolic notation, (2) *weakly verbalised* (e.g. “a is the limit of $(a_n)_{n\in\mathbb{N}}$”, “There exists a $\varepsilon > 0$ for which holds that $\forall n_0(\varepsilon).\exists n,m \geq n_0(\varepsilon)\ldots$”) where some relations or quantifiers are partly verbalised, and (3) *strongly verbalised* (e.g. “For all ε holds: there exists a $n_0(\varepsilon) \in \mathbb{N}$ with $\ldots$”, “For $x < 1$, x^n converges on 0.”) where all relations and quantifiers are fully verbalised. Figure 5 shows the formula verbalisation grade for every exercise. An exercise is classified as weakly/strongly verbalised if at least one weakly/strongly verbalised formula sentence occurs. The analysis reveals that

- services for symbolic formulas (e.g. notation check) are worth development,
- being able to fully parse/render symbolic formulas and weakly verbalised formulas gives a coverage of at least 70%,
- a linguistic ontology is required to deal with strongly verbalised formulas.

5.2 Style of Sentences

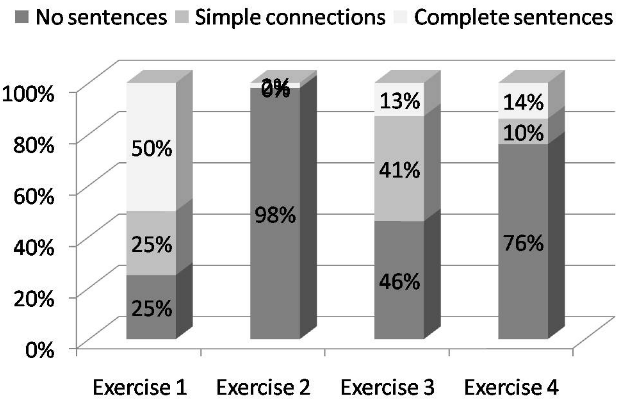


Figure 6: Style of Sentences

The style of sentences is classified into the following categories: (1) *no sentences* if there are no sentences at all, (2) *simple connections* if the sentences are built with simple key phrases (e.g. “Therefore it holds that ...”, “It holds ...”, “Hence we have ...”, “Thus it follows ...”, “Then ...”, “This means ...”, “Let ...”, “Select ...”, “Assuming ...”) and the building blocks are symbolic or weakly verbalised formulas, and (3) *complete sentences* if the sentences are fully verbalised with ontological building blocks like type, identifier, concept or attribute. The analysis in Figure 6 shows that

- a linguistic ontology is absolutely necessary to support text-style proofs like the one in the first exercise because a coverage of less then 50% is not acceptable,
- for computational proofs with case distinctions the handling of simple connectors and weakly verbalised formulas gives an excellent coverage of at least 85%.

5.3 Concluding Step

The concluding step is divided into the following categories: (1) *not present* if there is no step concluding the proof, (2) *symbolic* if there is a ■, "qed" or "q.e.d." at the end of the proof, and (3) *verbalised* if the concluding step is a fully verbalised sentence (e.g. "Therefore $(a_n)_{n \in \mathbb{N}}$ is a Cauchy sequence.", "... hence the theorem holds.", "This contradicts the assumption.", "... and thus $|a_n - a_m| < \varepsilon \quad \forall n, m \geq n_0(\varepsilon)$"). Note that there are two types of verbalised concluding steps, one *abstract* type that could be used for every proof of the same style, and one *concrete* type that performs the last inference, repeats the theorem or justifies the last conclusion. Figure 7 shows the analysis of concluding steps which draws the following conclusions:

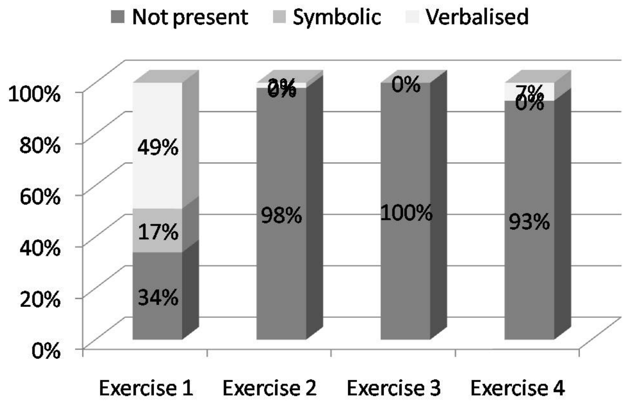


Figure 7: Concluding Step

- Verbalised concluding steps do almost never occur in computational proofs, even symbolic conclusions are very rare.
- In correlation to the sentence style aspect we can observe that 50% of the text-style proofs require a linguistic ontology in order to understand the verbalised concluding step.

5.4 Justifications

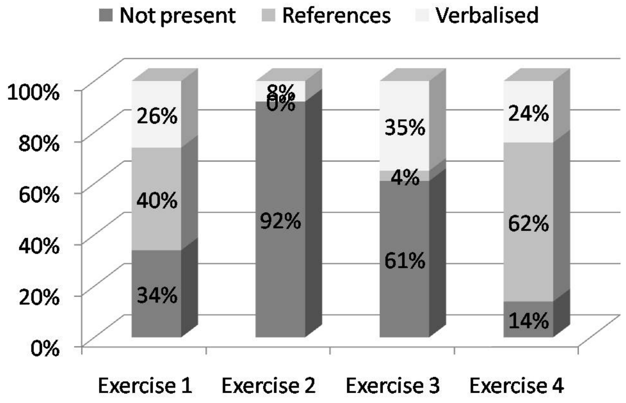


Figure 8: Justifications

The last aspect concerns the justification of proofsteps and consists of the following categories: (1) *not present* if no proofstep is justified at all, (2) *references* if the proofsteps are justified by referring to used theorems or concepts (e.g. "... with the triangle inequality", "From 14.23 it holds with $p = -1$ ...", "$x_1 \xRightarrow{\text{Theorem 14.23}} x_2$"), and (3) *verbalised* if the justification is a fully verbalised sentence possibly with references to proof strategies, complex methods or other sources of knowledge (e.g. "Since ... it holds ...", "... holds because ... and ...", "By polynomial division we compute ...", "Reason: Main coefficient is -1", "In the lecture we proved that ...", "From the first part of the exercise it follows that ...", "It's a matter of common knowledge that ..."). An exercise is classified as references/verbalised if at least one reference/verbalised justification sentence occurs. The analysis in Figure 8 reveals that

- not supporting verbalised references results in a medium coverage of at most 65% in the worst case, which might result in a low user acceptance,
- different linguistic ontologies are required to effectively understand verbalised references: a *document* ontology (e.g. for references to other parts of a document), an *argumentation* ontology (e.g. for references to the lecture or domain-specific techniques), and a *concept* ontology (e.g. for references to definitions). At least the concept ontology needs to be extended dynamically for example whenever new definitions or named local hypothesis are stated.

6 Discussion

In order to develop adequate tools for the interactive authoring of mathematical documents, we first of all have to choose the most suitable linguistic techniques for our scenario among the available shallow and deep methods. Therefore we conducted an exploratory study to identify the relevant linguistic phenomena that need to be supported by an ideal mathematical assistance system.

6.1 Lessons Learned

Dynamic documents as interfaces in text-editors are an interesting alternative to AJAX-driven web interfaces. Regarding the edit behaviour we observed a monotonic increasing with frequent use of copy & paste. The last sentence is quite often modified locally, global refactoring is used very rarely and only declaratively. Notational consistency checks would be very helpful because a lot of modifications were related to notational errors. Regarding the linguistic aspects we analysed that being able to fully deal with symbolic and weakly verbalised formulas, simple connected sentences and references leads to an average coverage of 70% in our scenario. This level might be reached with generic methods. By integrating linguistic ontologies (document, argumentation, concept) and mechanisms to extend these ontologies dynamically, we might reach a coverage of more than 90%.

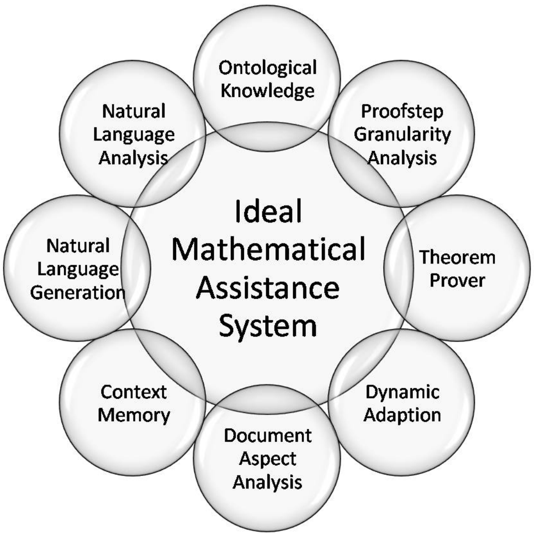

Ontological Knowledge
Natural Language Analysis
Proofstep Granularity Analysis
Ideal Mathematical Assistance System
Natural Language Generation
Theorem Prover
Context Memory
Dynamic Adaption
Document Aspect Analysis

Figure 9: Components of an IMAS

6.2 Ideal Mathematical Assistance System

According to the results of our study, an ideal mathematical assistance system (IMAS) should have at least the components shown in Figure 9:

- *Natural Language Analysis and Generation*: Weakly or strongly verbalised sentences have to be analysed resp. generated. We are currently integrating the shallow natural language generation system TG/2 [5] for this purpose.
- *Context Memory*: Natural language has to be generated in adaption of the style used in the context.
- *Document Aspect Analysis and Dynamic Adaption*: Different linguistic phenomena have to be analysed and put into the context, notation and references should be dynamically adapted. We will use the mediator system PLATΩ [3] for incremental parsing/rendering and semantic-based refactoring.
- *Theorem Prover and Proofstep Granularity Analysis*: The job of the theorem prover is to verify the correctness of the proofs or to generate missing proofsteps at an appropriate level of granularity. We will use the theorem prover ΩMEGA [10] as sophisticated foundation.
- *Ontological Knowledge*: In order to deal with strongly verbalised sentences we need to build up and dynamically extend different linguistic ontologies.

Zinn's computational framework[12] and the linguistic analysis of mathematical textbook proofs in his PhD thesis provide valuable insight into building linguistic ontologies for this purpose. Furthermore Zinn discusses the problems arising during the linguistic analysis of informal mathematical discourse.

7 Conclusion and Outlook

By analysing the corpus collected in an exploratory study with a first year mathematical course we were able to identify some design requirements for an authoring support system. The results of the conducted study are of course specific to first year mathematics students, we need further studies to transfer the results reliably to the general case. Due to the positive feedback of the lecturer and students we continue the exploratory study in the subsequent mathematical course. Beside that we plan to build two prototype systems:

- a *generic* system with a focus on scalability that supports weakly verbalised expressions and justifications by reference,
- a *specialized* system with ontological encoding of the domain knowledge and corresponding lexicon and grammar for natural language analysis and generation.

To reasonably limit the amount of work that needs to be invested into the second system, we choose a basic domain like *Simple Set Theory*. The aim is to evaluate and compare both systems with respect to user acceptance and return-on-invest.

Acknowledgments

The authors would like to thank Prof. Dr. Volker John, Michael Roland and their student assistants from the mathematics department of the Saarland University for supporting our study in the first year course *Mathematik für Informatiker*. This research is supported by the German National Academic Foundation.

References

[1] Adobe Systems Inc. *PostScript Language Reference, Third Edition*. Addison-Wesley, Boston, 1999.

[2] Adobe Systems Inc. *PDF Reference, Fifth Edition, Adobe Portable Document Format*. Adobe Press, 2005.

[3] S. Autexier, A. Fiedler, T. Neumann, and M. Wagner. Supporting user-defined notations when integrating scientific text-editors with proof assistance systems. In M. Kauers, M. Kerber, R. Miner, and W. Windsteiger, editors, *Towards Mechanized Mathematical Assistants*, LNAI. Springer, June 2007.

[4] Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions*. EATCS Texts in Theoretical Computer Science. Springer, 2004.

[5] S. Busemann. Ten years after: An update on TG/2 (and friends). In *Proceedings 10th European Workshop on Natural Language Generation*, August 2005.

[6] J. J. Garrett. AJAX: A new approach to web applications, 2005.

[7] D. E. Knuth. *The TEXbook*. Addison Wesley, 1984.

[8] L. Lamport. *LATEX: A Document Preparation System, 2/e*. Addison Wesley, 1994.

[9] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. LNCS 2283. Springer, 2002.

[10] J. Siekmann, C. Benzmüller, V. Brezhnev, L. Cheikhrouhou A. Fiedler, A. Franke, H. Horacek, M. Kohlhase, A. Meier E. Melis, M. Moschner, I. Normann, M. Pollet, V. Sorge, C. Ullrich, C.-P. Wirth, and J. Zimmer. Proof development with ΩMEGA. pages 143–148. Springer, 2002.

[11] J. van der Hoeven. Gnu TEXMACS: A free, structured, WYSIWYG and technical text-editor. Number 39-40 in Cahiers GUTenberg, May 2001.

[12] C. Zinn. Computational Framework For Understanding Mathematical Discourse. *Logic Journal of the IGPL*, 11(4):457–484, 2003.

Appendix

In the following we present the exercise sheet and some selected student solutions from our study, translated from German into English. The original material is available at `www.imath.eu` on request.

Exercise Sheet

1. Let $(a_n)_{n\in\mathbb{N}}$ be a convergent sequence in $\mathbb{R}$ with limit $a \in \mathbb{R}$. Show that it is a Cauchy sequence.
2. Compute the following limits of complex-valued sequences: $\lim_{n\to\infty} \frac{2in^3-n^4}{n^4+3in^2-1}$, $\lim_{n\to\infty} \frac{3n^2-2}{n^3+1}$, $\lim_{n\to\infty} \frac{5n-7n^2}{(n+1)^2-8n}$.
3. For $x \in \mathbb{R}_+$ compute $lim_{n\to\infty} \frac{x^n-n}{x^n+n}$.
4. Compute the following limits of the sequences $(x_n)_{n\in\mathbb{N}}$ with the help of limits proven in the lecture: $x_n = (1-\frac{1}{n})^n$, $x_n = (1-\frac{1}{n^2})^n$.

Solutions for Exercise 1

Requirement: $(a_n)_{n\in\mathbb{N}}$ is a convergent sequence in $\mathbb{R}$ with limit $a \in \mathbb{R}$.
Assume this sequence is not a Cauchy sequence. Then we have $\exists\varepsilon > 0$ such that for all $n_0(\varepsilon) \in \mathbb{N}$ holds: $\exists n$ and $m \geq n_0(\varepsilon) \quad |a_n - a_m| \geq \varepsilon$. Because a_n is a convergent sequence, it follows that: For all ε_2 it holds: there exists a $m_0(\varepsilon_2) \in \mathbb{N}$ with $|a_j - a| < \varepsilon_2 \quad \forall j \geq m_0(\varepsilon_2)$. Choose $\varepsilon_2 < \frac{\varepsilon}{2}$. Then we have: $|a_n - a| < \varepsilon_2$ and $|a_m - a| < \varepsilon_2$. But then it follows $|a_n - a_m| < 2\varepsilon_2$. Because of $\varepsilon_2 < \frac{\varepsilon}{2}$ it holds that: $|a_n - a_m| < 2\frac{\varepsilon}{2} = \varepsilon$. This is in contrast to the assumption. Hence a_n is a Cauchy sequence.

Given: $(a_n)_{n\in\mathbb{N}}$ is a convergent sequence in $\mathbb{R}$ with limit $a \in \mathbb{R} \Rightarrow \lim_{n\to\infty} a_n = a$
For each $\varepsilon > 0$ there exists a n_0 such that for all $n \geq n_0$ it holds that: $|a - a_n| < \frac{\varepsilon}{2}$.
$\forall n, m \geq n_0$ it follows by the triangular inequality:

$$|a_n - a_m| \leq |a_n - a| + |a - a_m| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

$\Rightarrow (a_n)_{n\in\mathbb{N}}$ is a Cauchy sequence.

Def. convergent sequence:
$\forall\varepsilon > 0 \exists n_0 : \forall n > n_0 : |a_n - a| < \varepsilon'$, additionally
$\forall m > n_0 : |a_m - a| < \varepsilon'$
By addition of both inequalities we get:
$|a_n - a| + |a_m - a| < 2\varepsilon'$
With the triangular inequality it follows by estimation:
$|a_n - a - a_m + a| < 2\varepsilon'$
$|a_n - a_m| < 2\varepsilon'$
If we rename $2\varepsilon'$ into ε, we get the theorem.

Solutions for Exercise 2

a) Let $x = \frac{2in^3-n^4}{n^4+3in^2-1} = \frac{2i\frac{1}{n}-1}{1+3i\frac{1}{n^2}-\frac{1}{n^4}} \Rightarrow \lim_{n\to\infty} x = \frac{0-1}{1+0} = -1$

b) Let $x = \frac{3n^2-2}{n^3+1} = \frac{3\frac{1}{n}-\frac{2}{n^3}}{1+\frac{1}{n^3}} \Rightarrow \lim_{n\to\infty} x = \frac{0}{1} = 0$

c) Let $x = \frac{5n-7n^2}{(n+1)^2-8n} = \frac{5n-7n^2}{n^2+2n-8n+1} = \frac{5n-7n^2}{n^2-6n+1} = \frac{-7+5\frac{1}{n}}{1-6\frac{1}{n}+\frac{1}{n^2}} \Rightarrow \lim_{n\to\infty} x = \frac{-7+0}{1-0+0} = -7$

---

a) $\lim_{n\to\infty} \frac{2in^3-n^4}{n^4+3in^2-1} = \lim_{n\to\infty} \frac{-1+2\frac{i}{n}}{1+3i\frac{1}{n^2}-\frac{1}{n^4}} = \frac{-1+0}{1+0-0} = -1$

b) $\lim_{n\to\infty} \frac{3n^2-2}{n^3+1} = \lim_{n\to\infty} \frac{\frac{3}{n}-\frac{2}{n^3}}{1+\frac{1}{n^3}} = \frac{0-0}{1+0} = 0$

c) $\lim_{n\to\infty} \frac{5n-7n^2}{(n+1)^2-8n} = \lim_{n\to\infty} \frac{5n-7n^2}{n^2-6n+2} = \lim_{n\to\infty} \frac{-7+\frac{5}{n}}{1-\frac{6}{n}+\frac{2}{n^2}} = \frac{-7+0}{1-0+0} = -7$

Solutions for Exercise 3

For $x \in]0,1|$ we get the following limit:

$$\lim_{n\to\infty} \frac{x^n-n}{x^n+n} = \frac{\lim_{n\to\infty} x^n - n}{\lim_{n\to\infty} x^n + n} = \frac{0-1}{0+1} = -1$$

For $x = 1$ we get the following limit:

$$\lim_{n\to\infty} \frac{1^n-n}{1^n+n} = \lim_{n\to\infty} \frac{1-n}{1+n} = -1$$

For $x > 1$ we get the following limit:

$$\lim_{n\to\infty} \frac{x^n-n}{x^n+n} = \lim_{n\to\infty} \frac{1-\frac{n}{x^n}}{1+\frac{n}{x^n}} = \frac{1-0}{1+0} = 1$$

Case distinction for $x \in \mathbb{R}$:

1. $x < 1$: $\lim_{n\to\infty}(\frac{x^n}{x^n+n} - \frac{n}{x^n+n}) = \lim_{n\to\infty}(\frac{0}{0+n} - \frac{n}{0+n}) = 0 - 1 = -1$
2. $x = 1$: $\lim_{n\to\infty}(\frac{x^n}{x^n+n} - \frac{n}{x^n+n}) = \lim_{n\to\infty}(\frac{1}{1+n} - \frac{n}{1+n}) = 0 - 1 = -1$
3. $x > 1$: $\lim_{n\to\infty}(\frac{x^n}{x^n+n} - \frac{n}{x^n+n}) = 1 - 0 = 1$, because x^n grows faster than n.

Solutions for Exercise 4

$x_n = (1-\frac{1}{n})^n$
With Lemma 14.23 we have: $\lim_{n\to\infty}(1+\frac{p}{n})^n = e^p$
Let $p = -1$, then it holds $\lim_{n\to\infty}(1-\frac{1}{n})^n = e^{-1}$
$x_n = (1-\frac{1}{n^2})^n$
With $p = -\frac{1}{n}$ and Lemma 14.23 it follows: $\lim_{n\to\infty}(1+\frac{-\frac{1}{n}}{n})^n = \lim_{n\to\infty}(1-\frac{1}{n^2})^2 = e^{-n}$

---

for $x_n = (1-\frac{1}{n})^n$:
from Lemma 14.23 we know: $\lim_{n\to\infty}(1+\frac{p}{n})^n = e^p$
it follows with $p = -1$ that the limit of the sequence is e^{-1}.
for $x_n = (1-\frac{1}{n^2})^n$:
because $(1+\frac{1}{n^2})^n = ((1-\frac{1}{n})(1+\frac{1}{n}))^n$ it follows from the first part of the exercise that $\lim_{n\to\infty}(1-\frac{1}{n})^n = e^{-1}$ holds and with Lemma 14.23 additionally $\lim_{n\to\infty}(1+\frac{1}{n})^n = e^1$. Hence the limit of the sequence is $e^1e^{-1} = e^0 = 1$.