DiaWOz-II – A Tool for Wizard-of-Oz Experiments in Mathematics[⋆]

Christoph Benzmüller[1], Helmut Horacek[1], Ivana Kruijff-Korbayová[2], Henri Lesourd[1], Marvin Schiller[1], and Magdalena Wolska[2]

[1] Dept. of Computer Science
[2] Dept. of Computational Linguistics
Saarland University, Germany
{chris,horacek,henri,schiller}@ags.uni-sb.de,
{korbay,magda}@coli.uni-sb.de

Abstract. We present DiaWOz-II, a configurable software environment for Wizard-of-Oz studies in mathematics and engineering. Its interface is based on a structural *wysiwyg* editor which allows the input of complex mathematical formulae. This allows the collection of dialog corpora consisting of natural language interleaved with non-trivial mathematical expressions, which is not offered by other Wizard-of-Oz tools in the field. We illustrate the application of DiaWOz-II in an empirical study on tutorial dialogs about mathematical proofs, summarize our experience with DiaWOz-II and briefly present some preliminary observations on the collected dialogs.

Keywords: Dialog systems, natural language dialog in mathematics, tutoring systems, Wizard-of-Oz experiments.

1 Introduction

For the development of natural language dialog systems, experiments in the Wizard-of-Oz (WOZ) paradigm are a valued source of dialog corpora.[1]

Existing environments for WOZ experiments, even those for the domain of mathematics tutoring, generally operate in domains that either require only simple mathematical formulae (with operators like + and ×), or they separate the mathematical objects (geometric figures or equations) from the tutorial dialog (such as in the Wooz tutor [2], for example). In this paper we present our WOZ environment DiaWOz-II which, in contrast to that, enables the collection of dialogs where natural language text is interleaved with mathematical notation, as is typical for (informal) mathematical proofs. The interface components

[⋆] This work has been funded by the DFG Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB 378 (`http://www.coli.uni-saarland.de/projects/sfb378/`).

[1] A Wizard-of-Oz experiment [1] serves to test the usability of a hypothetical software system. The system is (partially) simulated by a human expert, the *wizard*. Typically, a mediator software partially implements the functionality of the simulated system.

of DiaWOz-II are based on the *what-you-see-is-what-you-get* scientific text editor $\mathrm{T_EX_{MACS}}$[2] [3]. DiaWOz-II provides one interaction window for the user and one for the wizard, together with additional windows displaying instructions and domain material for the user, and additional notes and pre-formulated text fragments for the wizard. All of these windows allow for copying freely from one to the other. Furthermore, our DiaWOz-II allows the wizard to annotate user dialog turns with their categorization. DiaWOz-II is also connected to a spell-checker for checking both the user's and the wizard's utterances.

This paper is organized as follows: In Sect. 2 we motivate the design of our system. In Sect. 3.1 we describe the $\mathrm{T_EX_{MACS}}$ *wysiwyg* editor, on which the interface of DiaWOz-II is based. The DiaWOz-II system is discussed in detail in Sect. 3. In Sect. 4 we discuss the application of DiaWOz-II in a recently completed series of experiments. Section 5 concludes the paper.

2 Design Aspects

General Requirements for WOZ Tools. We list some general requirements we considered in the development of DiaWOz-II:

Plausibility and Comfort. For WOZ experiments, it is crucial to maintain the user's belief that he is interacting with a fully artificial system. Therefore, the software that mediates between wizard and student should enable the wizard to conceal his human identity. This is not a trivial pursuit, since it is common sense that "people are flexible, computers are rigid (or consistent), people are slow at typewriting, computer output is fast" [4]. Thus, the WOZ tool is required to enable the wizard to respond to the participant quickly and comfortably and in a plausible way.

Suitability for Book-keeping. The main goal of WOZ experiments is the analysis of the interactions between the subjects and the simulated system. Therefore, the WOZ tool is required to record the dialogs using a representation format suitable for further processing and analysis.

Flexibility and Simplicity. The WOZ tool should be easily adjustable, so that it can be used under different experimental conditions and in different domains. Adjustments to the software should not significantly add to the complexity of carrying out a series of experiments, a process which by itself poses enough challenges.

Tool Integration. The WOZ tool should support the integration of other software components, for example, modules that already realize single parts of the simulated overall system.

Specific Requirements for DiaWOz-II. DiaWOz-II has been developed for application in the DIALOG project [5], which investigates the use of natural language dialog for teaching mathematical proofs. The particular research foci of the DIALOG project are natural language analysis, domain reasoning for mathematics, and tutorial aspects of mathematics tutoring.

[2] `www.texmacs.org`

In 2003, we carried out a first empirical study [6] in the WOZ paradigm in which we collected a corpus of tutorial dialogs on mathematical proofs in German. The study concentrated on the comparison between three tutoring strategies, namely the *Socratic*, *didactic* and the *minimal feedback* strategies. For this purpose, we developed the DiaWoZ [7] environment, the predecessor of DiaWOz-II. DiaWoZ supports complex dialog specifications, which were needed in order to specify a particular hinting algorithm used in the *Socratic* tutoring condition. DiaWoZ allows keyboard-to-keyboard interaction between the wizard and the student. The interfaces consist mainly of a text window with the dialog history and a menu bar providing mathematical symbols. Furthermore, the wizard can assign dialog state transitions and speech act categories to student turns w.r.t. the underlying dialog model. The DiaWoZ interface allowed free mixing of natural language text with mathematical symbols. Still, there was room for improvement with respect to the *plausibility and comfort* criterion postulated above. For example, the experiment participants suggested the use of the keyboard instead of the mouse for inserting mathematical symbols.

The first study motivated a second series of experiments [8], which we briefly describe in Sect. 4. In contrast to the first study, the more recent study imposes less constraints on the wizards' tutoring and assumes a rather simple dialog model. However, in comparison to the first study, the second study is more focused on linguistic phenomena and mathematical domain reasoning in tutorial dialogs and the interplay between these two.

Related Work. A variety of WOZ tools and dialog system toolkits already exist. Examples are the simulation environment ARNE [4], the SUEDE prototyping tool for speech user interfaces [9] and MD-WOZ [10].

In the domain of mathematics, a WOZ simulation of the ALPS environment [11] and the Wooz tutor [2] should be mentioned. In the case of ALPS, the Synthetic Interview (SI) method is used, i.e. the student formulates free-form questions in a chat window, and receives a video clip with an answer. In the ALPS system, these video clips are pre-recorded, stored in a database, and retrieved as answers for the questions from the user, whereas in the WOZ simulation of ALPS, the wizard's responses are spontaneous. The ALPS tutor is designed to be an algebra tutor. Typical problems in the domain of ALPS are for example the computation of area and perimeter of geometric figures.

The Wooz tutor is also a tool for keyboard-to-keyboard interaction in the domain of algebra. It offers a chat window displaying the tutorial dialog, a dedicated window displaying the problem statement and a dedicated editor for editing equations. A typical problem given to the participants is "please factor $11x^2 - 11x + 6$".

The interfaces of these two systems are not intended for mixing natural language input with the mathematical notation employed for proving theorems, which we investigate in the DIALOG project. For our dialog system we aim for an interface that allows flexible and easy input for mathematical formulae and natural language text. This requirement is addressed by the interface in DiaWOz-II.

3 The DiaWOz-II System

We decided to build a new WOZ tool rather than trying to improve the existing DiaWoZ system. An important motivation was to use TeXmacs [3] as a platform for the new system in order to benefit from its typesetting abilities, its configurable GUI and its event-handling as a building block for the creation of a more lightweight software.

DiaWOz-II is realized as a classical client-server architecture, and consists of a server and two client interfaces for the student and the tutor respectively. The architecture allows keyboard-to-keyboard interaction between the student and the tutor. Furthermore, the server fulfills other central functions, namely the recording of the interaction in a log-file, controlling turn-taking between the dialog participants, and providing an interface to a spell-checker. We first describe TeXmacs and its role in DiaWOz-II before we further elaborate on each of these aspects in turn.

3.1 TeXmacs

TeXmacs is a scientific text editor with strong support for mathematical typesetting which is inspired by TeX and *GNU emacs*. The internal representation of a TeXmacs-document is well organized in a tree-like structure. TeXmacs provides two alternative editing modes: (i) a *wysiwyg interface* that allows to directly manipulate the typeset document and (ii) a *source mode* that provides a view of the internal document representation in the underlying, structured TeXmacs markup language. This language supports the definition of *macros*, which are generally easy to read and understand. It is also worth noting that the standard TeXmacs markup language inherits many usual LaTeX constructs, in such a way that for LaTeX-literate persons, starting to use TeXmacs is usually straightforward. Thus, extending the markup (namely, defining new kinds of tags together with how these newly defined tags must be typeset) can be done in a very convenient way using macros. For more sophisticated behavior, for example, the implementation of an interactive application, one can use *Scheme*, the standard TeXmacs scripting language.

TeXmacs fulfills the *plausibility and comfort* requirement introduced in Sect. 2 by offering various advanced modes of input for mathematical symbols, and in particular it enables LaTeX commands. Using TeXmacs also fulfills the *flexibility and simplicity* requirement, since it can be reconfigured with little effort.

The TeXmacs editor has already been adapted as an interface to a diversity of external tools, most of which are computer algebra systems. However, using TeXmacs as an interface for a (simulated) tutoring system is novel.

3.2 TeXmacs as Base Component of DiaWOz-II

A TeXmacs application as employed in DiaWOz-II has the overall structure shown in Fig. 1. Such an application consists of (i) a set of TeXmacs *macros* which implement the *visualization* of the different parts of the user interface (i.e. what are their shapes, their locations, the text attributes (e.g. color, font, ...), etc.), and

(2) a set of *Scheme scripts*, which implement the mechanism which interprets the *events* (e.g., a mouse click, a key press, etc.) and *modifies* the interface accordingly.

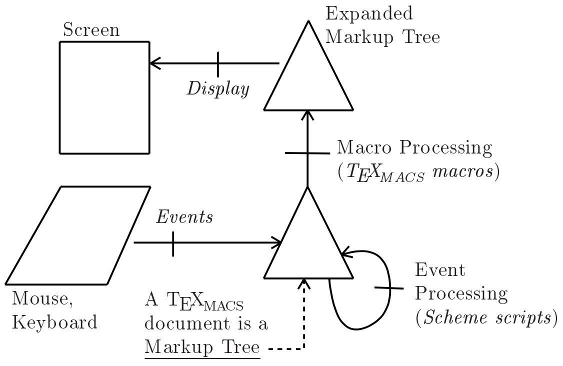

Screen
Expanded Markup Tree
Display
Macro Processing (T_EX_{MACS} *macros*)
Events
Event Processing (*Scheme scripts*)
Mouse, Keyboard
A TEXMACS document is a Markup Tree

Fig. 1. Structure of a TEXMACS application

Macros. A very basic example of a TEXMACS macro that can be used to turn a part of the document into *italics underlined* text is (cf. [12] for more details on the macro language):

```
<underlined-italics|x> => <with|font-shape|italic|<underline|<arg|x>>>
```

The left-hand side of this expression defines the use of the macro (i.e., the *non-expanded* markup, as it can be found in a TEXMACS document file) and the right-hand side its expansion. Given this macro definition, the TEXMACS markup fragment `<underlined-italics|This is italics underlined text.>` is first rewritten by the macro processor as `<with|font-shape|italic| <underline|This is italics underlined text.>>` and then displayed in TEXMACS as *This is italics underlined text.*

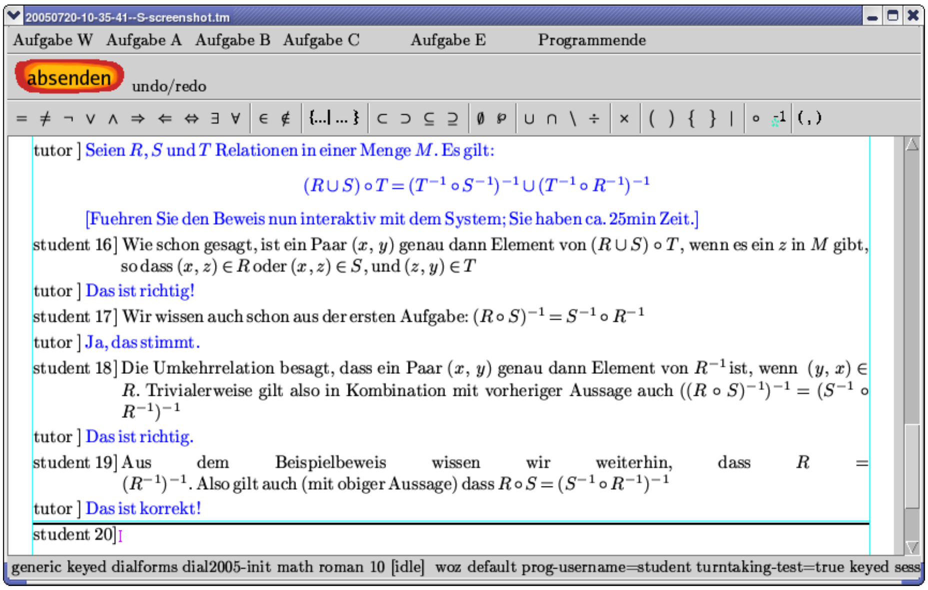
Processing the Markup Using *Scheme*. The event processor can be extended by plugins written as *Scheme* scripts. These scripts can manipulate the internal markup tree that represents the user interface, typically as a reaction to an event (e.g., mouse, keyboard, network, etc.). As a reaction to the changes in the markup, the macros are reevaluated, and the display is then updated.

3.3 Student and Wizard Interfaces

The dialog system simulated by DiaWOz-II is presented to the student as a window, referred to as the interaction window. It consists of menu bars and a text field, as shown in Fig. 2. The dialog history and the prompt for the current input are displayed in the same text field, separated by a horizontal bar at the bottom in Fig. 2. The utterances from the tutor and the student are displayed in different colors for better readability. The student can send messages by pressing the "absenden" (submit) button. Upon submitting, the message becomes part of the dialog history. The answers by the tutor are accompanied by an acoustic signal.

In a second window, which is independent of the interaction window, supplementary study material with mathematical concepts and definitions is displayed.

The wizard's interface, as shown in Fig. 3, is conceptually similar to the student's interface. In addition, the wizard is asked to categorize each student turn w.r.t. three dimensions: correctness, granularity and relevance; the wizard fills out the fields of a small table referring to the three dimensions by making


20050720-10-35-41--S-screenshot.tm
Aufgabe W
Aufgabe A
Aufgabe B
Aufgabe C
Aufgabe E
Programmende
absenden
undo/redo
$= \neq \neg \vee \wedge \Rightarrow \Leftarrow \Leftrightarrow \exists \forall \in \notin \{..|...\} \subset \supset \subseteq \supseteq \emptyset \wp \cup \cap \setminus \div \times ( ) \{ \} | \circ {}^{-1} (,)$

tutor] Seien R, S und T Relationen in einer Menge M. Es gilt:

$$(R \cup S) \circ T = (T^{-1} \circ S^{-1})^{-1} \cup (T^{-1} \circ R^{-1})^{-1}$$

[Fuehren Sie den Beweis nun interaktiv mit dem System; Sie haben ca. 25min Zeit.]

student 16] Wie schon gesagt, ist ein Paar (x, y) genau dann Element von $(R \cup S) \circ T$, wenn es ein z in M gibt, so dass $(x, z) \in R$ oder $(x, z) \in S$, und $(z, y) \in T$

tutor] Das ist richtig!

student 17] Wir wissen auch schon aus der ersten Aufgabe: $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$

tutor] Ja, das stimmt.

student 18] Die Umkehrrelation besagt, dass ein Paar (x, y) genau dann Element von R^{-1} ist, wenn $(y, x) \in R$. Trivialerweise gilt also in Kombination mit vorheriger Aussage auch $((R \circ S)^{-1})^{-1} = (S^{-1} \circ R^{-1})^{-1}$

tutor] Das ist richtig.

student 19] Aus dem Beispielbeweis wissen wir weiterhin, dass $R = (R^{-1})^{-1}$. Also gilt auch (mit obiger Aussage) dass $R \circ S = (S^{-1} \circ R^{-1})^{-1}$

tutor] Das ist korrekt!

student 20]

generic keyed dialforms dial2005-init math roman 10 [idle] woz default prog-username=student turntaking-test=true keyed sess

Fig. 2. Interaction window of the student interface

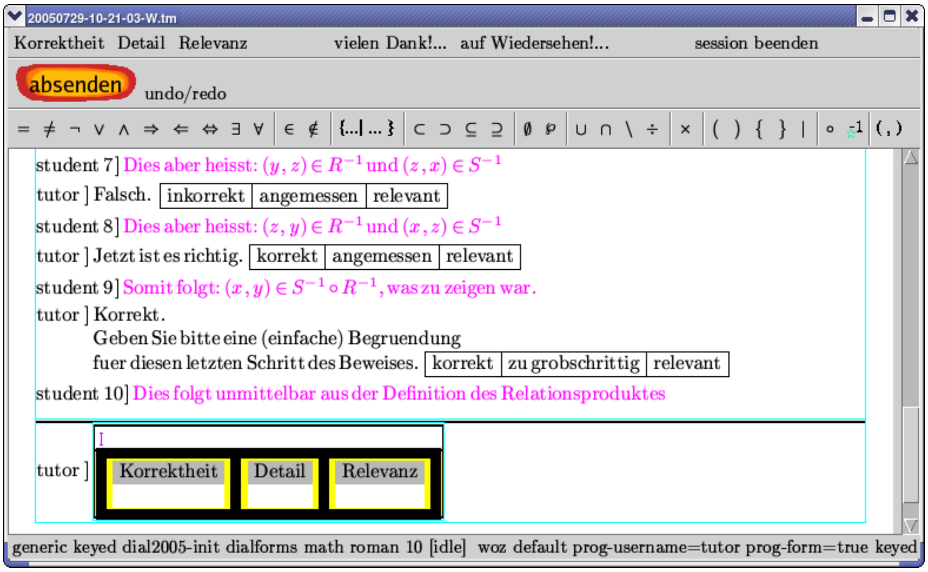
choices in pull-down menus, or directly by typing. The wizard's button for sending messages is only enabled once all the fields have been filled. If the student's utterance does not represent a mathematical statement the wizard fills in default values (`N/A`).

We now turn in more detail to the methods for inserting mathematical symbols in DiaWOz-II, which are made available by TEXMACS. Mathematical symbols (e.g., $\emptyset$) can be created by using LATEX commands (e.g., `\emptyset`) or by using additional commands defined when designing the interface (e.g., the command `\emptyset` in German language, i.e. `\leeremenge`). These commands are also made available in the menu bar. DiaWOz-II also allows for structured commands, e.g. commands that create pairs of brackets for pair $(\square, \square)$ and for set notation $\{\square | \square\}$. An example is the macro *paar* (German for *pair)*:

```
<paar|left|right> => ( <arg|left> , <arg|right> )
```

Invoking `\paar` with the arguments x and y yields the formula (x, y). The two arguments need not necessarily be provided when invoking the macro, their respective placeholders can also be filled in interactively and modified later. Macros can be nested, and most importantly, they avoid missing parentheses when the user writes expressions using the pair notation. The set of macros provided with DiaWOz-II can be easily extended with further TEXMACS macros.

TEXMACS furthermore makes it possible to distinguish between mathematical symbols created via the menu bar and via LATEX commands, even if they appear to be the same at the typesetting level.


20050729-10-21-03-W.tm
Korrektheit Detail Relevanz vielen Dank!... auf Wiedersehen!... session beenden
absenden
undo/redo
Korrektheit
Detail
Relevanz

student 7] Dies aber heisst: $(y,z) \in R^{-1}$ und $(z,x) \in S^{-1}$

tutor] Falsch. | inkorrekt | angemessen | relevant |

student 8] Dies aber heisst: $(z,y) \in R^{-1}$ und $(x,z) \in S^{-1}$

tutor] Jetzt ist es richtig. | korrekt | angemessen | relevant |

student 9] Somit folgt: $(x,y) \in S^{-1} \circ R^{-1}$, was zu zeigen war.

tutor] Korrekt.
Geben Sie bitte eine (einfache) Begruendung
fuer diesen letzten Schritt des Beweises. | korrekt | zu grobschrittig | relevant |

student 10] Dies folgt unmittelbar aus der Definition des Relationsproduktes

tutor]

```
generic keyed dial2005-init dialforms math roman 10 [idle]  woz default prog-username=tutor prog-form=true keyed
```

Fig. 3. The interaction window of the wizard interface

Using structured building blocks for constructing mathematical formulae via macros is similar to the Maths Tiles approach [13]. Maths Tiles are graphical tiles that can contain text, diagrammatic shapes and *sockets*, which are placeholders where other Maths Tiles can be inserted to form composite objects. TeXmacs has the advantage over Maths Tiles that it already includes by default a large set of macros for constructing formulae, such as a large number of macros that represent LaTeX commands.

3.4 The Server

The central capabilities of DiaWOz-II reside in the server. Its main task is to pass the dialog contributions back and forth between the student and the wizard interface. Furthermore, it provides the following other central services:

Log-file Mechanism. All dialogs are recorded in a log-file in DiaWOz-II. The log-file format is based on the representation format of TeXmacs, which is a structured, extensible and open document format. Naturally, the annotations performed by the wizard for each student turn are also stored in the log-file.

Spell-Checking. Spelling mistakes by the wizard can be a giveaway of human simulation. Therefore, our server (optionally) integrates a spell-checker. If spell-checking is activated, a message from the wizard is only passed on by the server if it passes the spell-checker, otherwise the wizard is asked to correct the message. The student's input is also spell-checked. Messages exceeding a threshold of spelling errors are refused (i.e. not passed on to the wizard). The underlying rationale is that it would be implausible that an automated system could deal with such misspelled input.

We currently employ the spell-checker GNU Aspell[3] with the standard German dictionary provided with Aspell together with an extra dictionary of mathematical jargon. The latter was compiled from the introductory mathematics materials and gradually extended during the experiments.

Turn-Taking Control. DiaWOz-II imposes strict turn-taking on the student: once the student makes a turn, the sending of new messages is disabled (i.e., the dedicated button for "sending" is deactivated and displayed in a darker shade) until the tutor provides a response. Without this constraint, it might become unclear to which turn of the student an answer from the wizard belongs. However, the tutor is allowed to barge in at any time, which enables him to offer support or prompt if the student appears to be inactive.

3.5 Implementation

Figure 4 illustrates the architecture of DiaWOz-II. In order to customize the client interfaces, we have

– adapted the menu bars and buttons to the needs of our application and
– restricted the editing facilities so that the student can only type in a designated text area with all other TEXMACS functionalities disabled (for example, inserting an image, or editing the dialog history).

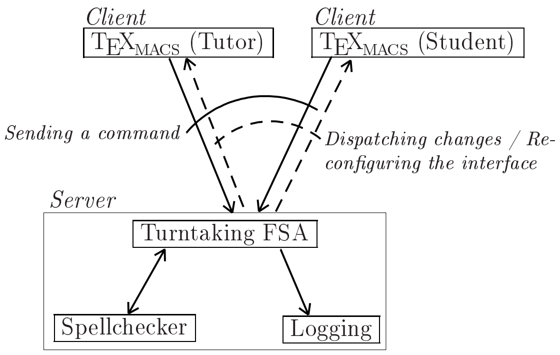


Fig. 4. System architecture

On the server side, turn-taking is controlled by a finite-state automaton. A message received by the server is written to the log-file and sent to a spell-checker. If it passes, it is broadcast to the clients. If it does not pass, it is sent back to the sender for correction. Disallowing the student from sending new messages until the wizard makes a turn is technically realized by server messages to the student's client to reconfigure the client's interface (i.e. enable/disable the interface's elements according to the current state).

The combination of macros and *Scheme* provided in TeXmacs has turned out to be very useful for our development of DiaWOz-II. In particular, the amount of code we wrote (a dozen of *Scheme* files of approximately 100 Kb in total) is relatively small considering the implemented functionality, and it remained manageable over time (as opposed to the previous version of DiaWoZ that consisted of about 200 Kb Java code spread among 70 files). The environment enabled also

[3] `http://aspell.sourceforge.net/`

people who are not professional software developers to participate in developing the system. Thus, TeXmacs has proven to be a good choice for our WOZ software, both from the point of view of the level of functionality it offers (word processing with LaTeX-like mathematical typesetting in a customizable editor) as well as from the point of view of prototyping and extending the software. The combination of the *Scheme* programming language with the large set of features already provided by TeXmacs allows for a lightweight, inclusive software development process.

4 An Empirical Study Using DiaWOz-II

Exploiting the DiaWOz-II system, we carried out a series of experiments in July 2005. In this study (see [8]), we collected a corpus of tutorial dialogs in German on mathematical proofs in the domain of binary relations. The collected data serves to investigate linguistic phenomena related to the mixing of mathematical formulae and natural language, underspecification phenomena, qualitative aspects of proof steps and mutual dependencies between natural language analysis and non-trivial mathematical domain reasoning.

4.1 Method

Thirty-seven students from Saarland University participated in the experiments. They were instructed to solve proof exercises collaboratively with a computer system that was described to them as a natural language dialog system on mathematics. This system was simulated with the DiaWOz-II software and four experts[4], who took the role of the wizard in turn (the set-up is shown in Fig. 5).

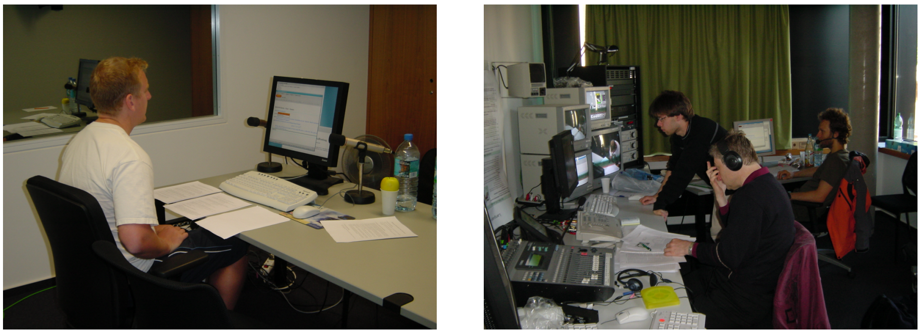


Fig. 5. An experiment in progress: The participant (left) and the wizard, experimenter and research assistant in the control room (right)

The wizards were given general instructions on the *Socratic* style of tutoring (cf. [14]), which is characterized by the use of questions to elicit information from

[4] The experts consisted of the lecturer of a course *Foundations of Mathematics*, a maths teacher, and two maths graduates with teaching experience.

the student. The tutors were instructed to reject utterances outside the mathematical domain and to respond in a uniform manner. Apart from that, the wizards were not restricted in the verbalization of their answers to the students. This allowed us to investigate the use of mathematical language without possibly influencing it by a-priori restrictions, even if more restrictions might have contributed to making the simulated system appear even more machine-like. In addition to the interaction window of DiaWOz-II, the tutors were provided with a second TeXmacs window in which they could save text and formulae for re-use.

The exercises were taken from the domain of relations, and were centered around the concepts of relation composition and relation inverse. Because of the advanced character of the exercises, the participants were required to have taken part in at least one mathematics course at university level. First, the subjects were required to fill out a questionnaire, asking about previous experiences with dialog systems and mathematics background. Subjects were also given study material with the mathematical definitions that were required to solve the exercises which was studied for approximately 25 minutes. The materials were handed out on paper and were also available as a TeXmacs document on the screen. This helped to achieve a uniform (and thus plausible) appearance of the system. Prior to the tutoring session, the students received a short introduction to the interface, during which the different modes of input for mathematical symbols – as menu items, as LaTeX commands or via commands in German language – and the copy & paste facility were demonstrated.

The largest part of the two-hour experimental session was allotted to the interaction between the student and the simulated system. In addition to the log-files recorded by DiaWOz-II, screen recordings were made. Furthermore, the participants were encouraged to "think aloud" and they were audio-recorded and filmed. This comprehensive collection of data not only documents the text of the tutorial dialogs, but also allows us to analyze how the participants used the interface and the study material.

At the end of the experimental session, the participants were required to fill out a second questionnaire asking about their verdict on the usability of the system, how difficult they found the exercises, and suggestions for improvements of the system.

4.2 Discussion

The experiments resulted in a large and diverse corpus of dialogs. During a session, a participant made on average 24 dialog turns, excluding those that were rejected for bad spelling. We briefly discuss how DiaWOz-II fulfilled its role, how the participants coped with the interface. Furthermore, the collected data hints at a potential influence of the interface features in combination with the reading material on the resulting tutorial dialogs.

Observations from the Corpus. An example of two dialog fragments from the experiment is given in Fig. 6. These dialogs were obtained under two different modes of presentation of the study material: formal (FM) vs. verbose (VM). Note

S33: Nach Aufgabe W ist $(\mathrm{S} \circ (\mathrm{S}{\cup}\mathrm{R})^{-1})^{-1} = ((\mathrm{S}{\cup}\mathrm{R})^{-1})^{-1} \circ \mathrm{S}^{-1}$
By Exercise W $(S{\circ}(S{\cup}R)^{-1})^{-1}{=}((S{\cup}R)^{-1})^{-1}{\circ}S^{-1}$ *holds*
T34: Das ist richtig! *That is correct!*
S34: Dies ist nach Theorem 1 gleich $(\mathrm{S}{\cup}\mathrm{R}) \circ \mathrm{S}^{-1}$
This is by Theorem 1 equal to $(S{\cup}R) \circ S^{-1}$
T35: Das ist auch richtig!
That is also correct!
S35: Ein Element (a,b) ist genau dann in dieser Menge, wenn es ein z∈M gibt mit (a,z)∈S∪R und $(z,b){\in}\mathrm{S}^{-1}$
An element (a,b) is in this set exactly when there is a z∈M with (a,z)∈S∪R und $(z,b){\in}S^{-1}$
T36: Das ist korrekt! *That is correct!*
S36: Also (b,z)∈S. *Therefore (b,z)∈S.*
T37: Auch das ist korrekt!
That is also correct!
S37: DAmit [sic] kann aber nicht die geforderte Gleichheit bewiesen werden.
But with this the given equality cannot be proven.
T38: Das haben Sie gut erkannt! Was fehlt Ihnen dazu?
You have made a good observation! What is missing?

S25: $(\mathrm{R}{\cup}\mathrm{S}){\circ}\mathrm{S} = \{(x,y) \mid \exists\, z(z \in M \wedge ((x,z) \in R \vee (x,z) \in S) \wedge (z,y) \in S\}$
T28: Diese Beziehung hat Relevanz fuer die Aufgabe. Sie muessen sich also etwas dabei gedacht haben, als Sie diese vermuteten. Koennen Sie begruenden, wie sie mit der Aufgabe zusammenhaengt?
This relation is relevant for the exercise. You must have considered something as you proposed it. Can you justify how it is related to the exercise?
T29: Ihre Formel ist richtig.
Your formula is correct.
T30: Koennen Sie die Behauptung unter zusaetzlichen Voraussetzungen beweisen?
Can you prove the expression under additional assumptions?
S26: $\{(x,y) \mid \exists\, z(z \in M \wedge ((x,z) \in R \vee (x,z) \in S) \wedge (z,y) \in S\} = \{(x,y) \mid \exists\, z(z \in M \wedge (y,z) \in S \wedge ((x,z) \in R \vee (x,z) \in S))\} \Leftrightarrow ((y,z) \in S \wedge (z,y) \in S)\}$
T31: Auf der rechten Seite ist z nicht spezifiziert. Meinten Sie vielleicht ⇔ oder etwas Aehnliches [sic] statt ∧?
On the right side z is not specified. Do you perhaps mean ⇔ or something similar instead of ∧?

Fig. 6. Excerpts of dialogs in the two conditions: VM-group (left) and FM-group (right). English translations are given in italics. S_ and T_ indicate student and wizard turns, respectively.

that the dialogs clearly differ in the employed mathematical style and that in Fig. 6 (right), the mathematical operations performed by the student can be characterized as term rewriting steps, i.e. a subformula of a term is replaced by an equivalent subformula. Also note that in Fig. 6 (right), the student uses no natural language. Even though all subjects were informed before the interaction that the system can handle a combination of natural language and formula input, we observed great variations in the amount of natural language used by the subjects.

Corpus analysis reveals differences in the use of natural language and mathematical expressions that was at least partially influenced by the mode of presentation of the study material. The group presented with the verbose material tended to use more natural language than the formal material group and the dialog turns of the VM-subjects contained more, but shorter, mathematical expressions. The formal material group tended to use more and longer formulas overall, and less natural language. More details on the differences in language production between the two conditions can be found in [15].

The copy & paste facilities provided by DiaWOz-II allowed copying definitions from the study material into the dialog contributions, and allowed copying previously uttered formulae for constructing new formulae. We observed that many subjects constructed larger and larger formulae with several levels of nesting. No such phenomenon was observed in the first study [6]. Even though the predecessor DiaWoZ software used in this study allowed copy & paste, this feature was not explained to the users and discovered only by some. Furthermore, in the first study the introduction material was only presented on paper, so that students could not copy from there as was possible in the second study. Another difference is the mathematical domain itself - the proofs concerning relations in the second experiment series require considerably longer formulae than those concerning naive set theory in the first experiment.

Usability of DiaWOz-II. The students were required to fill out post-experiment questionnaires, which among other things asked questions about the interface.

Students were asked if they had problems while using the interface, and to qualify their answer by a rating on a five-point scale between one (no problems) and five (great problems). Their ratings[5] (median 2, average 2.14, standard deviation 0.85) indicate that the participants generally had little trouble using the DiaWOz-II interface.

Even though a direct comparison between DiaWoZ and DiaWOz-II would require an experiment on its own (the two reported experiments involved different mathematical domains and different requirements imposed on the participants), these ratings are not far from those obtained in the first series of experiments with DiaWoZ. There, students had also been asked the same question, where they indicated a rating of 1.59 on average and a median of 1.

A small number of participants commented to the experimenter that they suspect a human teacher. However, comments by other subjects indicated that these were convinced of having interacted with an automated system.

Participants were asked to give comments about the system in general and the interface in particular, which are summarized in Table 1. The fact that the input facilities of DiaWOz-II were positively mentioned by numerous participants can be contrasted with the first series of experiments, where eight of the seventeen participants complained that the sole input method for mathematical symbols via the menu bar required constant switching between the mouse and the keyboard for inputting mathematical formulae.

A serious criticism concerned the speed of the system. This refers to two aspects: (1) the fact that the students had to wait for the answers from the

[5] The ratings from thirty-six participants are distributed as follows: A rating of 1 was assigned by 7 participants, a rating of 2 by 21 participants, a rating of 3 by 4 participants and a rating of 4 by 4 participants. No participant gave a rating of 5.

Table 1. Most frequent comments on the DiaWOz-II interface (number of participants indicated in brackets)

Positive Comments	
– Variety of formula input methods[1] (7)	– Interface is simple to use/clear (5)
– LaTeX commands available[1] (6)	– Questions can be formulated in NL (4)
– Math symbols in menu[1] (5)	
[1] In total, 20 subjects mentioned at least one positive aspect w.r.t. to formula input.	
Negative Comments	
– TeXmacs-specific problems (14)	– Interface delay (10)
– Bad screen size/font size (8)	– Sending messages not via return key (6)
– No direct keyboard shortcuts for math symbols available (3)	

system, and (2) the behavior of the interface itself. The waiting times consisted in the time spent by the tutor to read the dialog contributions from the students and to write an answer (even with the help of pre-formulated answers), but also the message-passing between the client, the server and the spell-checker. An important fact was that the wizards were sometimes challenged by the size of formulae created by the students, which made checking them particularly time-consuming. The insufficient speed attributed to the system's interface refers to a small but noticeable delay when typing symbols in DiaWOz-II. This delay is not experienced when using a standard TeXmacs, but results from the extra mechanism that protects the dialog history from being edited mentioned above. Another criticism concerns the window layout. For the experiment we used a screen capturing software and a low screen resolution to save disk space, which was commented on negatively by the subjects.

In summary, the questionnaires show that the input methods for mathematical text available in DiaWOz-II were well received by many participants, but that other mainly technical difficulties remain. A possible improvement proposed by some of the participants is an option for the user to withdraw a message after it is sent, in case the user himself becomes aware of a minor error and wants to correct it himself.

5 Conclusion

We have presented DiaWOz-II, our mediator software for WOZ experiments based on the *wysiwyg* editor TeXmacs. DiaWOz-II allows various modes of input for mathematical symbols, such as LaTeX commands, customized commands and menu items, and editing facilities that allow for the creation of complex formulae. Furthermore, DiaWOz-II inherits high quality typesetting from TeXmacs. One purpose of this paper is to advocate DiaWOz-II to the AI community for similar WOZ studies in domains such as engineering, physics, economics, etc. where mathematical input in combination with natural language plays a crucial role.

We also briefly addressed the set-up and some results of a series of experiments conducted with DiaWOz-II. The corpus we obtained is important to guide our research in the DIALOG project. It is currently being evaluated and can be obtained from `http://www.ags.uni-sb.de/~dialog` (see [8] for a preliminary analysis). We have observed that the capabilities of DiaWOz-II for editing and copying mathematical formulae introduced artifacts into some of the tutorial dialogs that we collected, which we did not observe in the previous, similar experiment. These manifest themselves in a term-rewriting style of proving mathematical theorems leading to unnecessarily large and nested formulae. This hints at the importance of incorporating didactic knowledge into tutoring systems in our field (as simulated by DiaWOz-II) which prevent students from abusing such a system's features in a technology-driven manner, and to help the students to use these features purposefully and with moderation.

As a part of our ongoing work, we are combining the dialog specification mechanism from DiaWoZ with the DiaWOz-II system to obtain an environment that reflects our expertise gained with both systems. The DiaWOz-II system can be downloaded from `http://www.ags.uni-sb.de/~dialog/diawoz2`.

Acknowledgments. We would like to thank all of the members of the Dialog team for their input and comments on initial drafts of this paper, and of course for their contributions to DiaWOz-II and the experiments.

References

1. Fraser, N.M., Gilbert, G.N.: Simulating speech systems. Computer Speech and Language (5) (1991) 81–99
2. Kim, J.H., Glass, M.: Evaluating dialogue schemata with the Wizard of Oz computer-assisted algebra tutor. In: Intelligent Tutoring Systems. (2004) 358–367
3. Hoeven, J.v.d.: GNU TeXmacs: A free, structured, wysiwyg and technical text editor. In Flipo, D., ed.: Le document au XXI-ième siècle. Volume 39-40., Metz (2001) 39–50 Actes du congrès GUTenberg.
4. Dahlbäck, N., Jönsson, A., Ahrenberg, L.: Wizard of Oz studies – Why and how. Knowledge-Based Systems **6**(4) (1993) 258–266
5. Benzmüller, C., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Pinkal, M., Siekmann, J., Tsovaltzi, D., Vo, B.Q., Wolska, M.: Tutorial dialogs on mathematical proofs. In: Proceedings of the IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems, Acapulco (2003) 12–22
6. Benzmüller, C., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Pinkal, M., Siekmann, J., Tsovaltzi, D., Vo, B.Q., Wolska, M.: A Wizard of Oz experiment for tutorial dialogues in mathematics. In: Proceedings of AI in Education (AIED 2003) Workshop on Advanced Technologies for Mathematics Education, Sydney, Australia (2003) 471–481
7. Fiedler, A., Gabsdil, M., Horacek, H.: A tool for supporting progressive refinement of wizard-of-oz experiments in natural language. In Lester, J.C., Vicari, R.M., Paraguaçu, F., eds.: Intelligent Tutoring Systems — 7th International Conference (ITS 2004). Number 3220 in LNCS, Springer (2004) 325–335

8. Benzmüller, C., Horacek, H., Lesourd, H., Kruijff-Korbayová, I., Schiller, M., Wolska, M.: A corpus of tutorial dialogs on theorem proving; the influence of the presentation of the study-material. In: Proceedings of International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, ELDA (2006) To Appear.
9. Klemmer, S.R., Sinha, A.K., Chen, J., Landay, J.A., Aboobaker, N., Wang, A.: Suede: a wizard of oz prototyping tool for speech user interfaces. In: UIST. (2000) 1–10
10. Munteanu, C., Boldea, M.: MDWOZ: A Wizard of Oz environment for dialog systems development. In: Proceedings 2nd International Conference on Language Resources and Evaluation, Athens, Greece (2000) 1579–82
11. Anthony, L., Corbett, A.T., Wagner, A.Z., Stevens, S.M., Koedinger, K.R.: Student question-asking patterns in an intelligent algebra tutor. In Lester, J.C., Vicari, R.M., Paraguaçu, F., eds.: Intelligent Tutoring Systems. Volume 3220 of Lecture Notes in Computer Science., Springer (2004) 455–467
12. Hoeven, J.v.d., et al.: The TeXmacs manual. http://www.texmacs.org/tmweb/manual/web-manual.en.html (1999-2006)
13. Billingsley, W., Robinson, P.: Towards an intelligent online textbook for discrete mathematics. In: Proceedings of the 2005 International Conference on Active Media Technology, Takamatsu, Japan (2005) 291 – 296
14. Rosé, C.P., Moore, J.D., VanLehn, K., Albritton, D.: A comparative evaluation of socratic versus didactic tutoring. In: 23rd Annual Conference of the Cognitive Science Society, Edinburgh, Scotland (2001)
15. Wolska, M., Kruijff-Korbayová, I.: Factors influencing input styles in tutoring systems: the case of the study-material presentation format. In: Proceedings of the ECAI-06 Workshop on Language-enabled Educational Technology. (2006) To Appear.