Project MI 3:

DIALOG:
Tutorial Dialog with an Assistance System for Mathematics

3.1 General Information About the Project MI 3

3.1.1 Topic

3.1.2 Discipline and Field

3.1.3 Directors

Funding period:	1 January 2002 – 31 December 2004
Period covered by report:	1 February 2001 – 31 January 2004
Project directors:	Prof. Dr. Manfred Pinkal Prof. Dr. Jörg Siekmann Dr. Christoph Benzmüller
Work address:	FB Computerlinguistik FB Informatik Universität des Saarlandes Postfach 15 11 50 66041 Saarbrücken
Phone numbers:	(0681) 302 – 4343 / 5275 / 4574
Fax number:	(0681) 302 – 4351 / 2235 / 2235
Email addresses:	pinkal@coli.uni-sb.de siekmann@ags.uni-sb.de chris@ags.uni-sb.de
Researchers:	Dr. Armin Fiedler, 01.02 - current (3 months unpaid leave) PD Dr. Helmut Horacek (Grundausstattung), 01.02 - 06.03 Dr. Quoc Bao Vo, 05.02 - 07.02 Dimitra Tsovaltzi, MSc, 01.03 - 12.03 Magdalena Wolska, MPhil, 01.03 - current Dr. Ivana Kruijff-Korbayová (Grundausstattung), 01.02 - current Elena Karagjosova (Grundaustattung), 01.03 - 12.03 PD Dr. Erica Melis (DFKI), 01.02 - current

3.2 State of the Art at the Time of the Proposal and Original Research Issues

Our goal in the DIALOG project was to develop a flexible dialog system for tutoring mathematical problem solving, in particular, theorem proving. We set out to investigate what requirements the flexible dialog paradigm poses on the components of a tutorial dialog system.

Empirical findings in the area of intelligent tutoring show that flexible natural language dialog supports active learning (Moore, 1993). In the DIALOG project, therefore, the focus has been on the development of solutions allowing flexible dialog. However, little is known about the use of natural language in dialog settings in formal domains, such as mathematics, due to the lack of empirical data. To fill this gap, we designed and performed an experiment with a simulated tutorial dialog system for teaching proofs in naive set theory. An investigation of the collected corpus revealed challenging phenomena at all levels of system design: from input analysis, through mathematical domain reasoning, to tutorial dialog strategies.

Input analysis in dialog systems is for most domains commonly performed using shallow syntactic analysis combined with keyword spotting; slot-filling templates, however, are not suitable in our case. Moreover, tight interleaving of natural and symbolic language makes key-phrase spotting difficult because of the variety of possible verbalizations. Statistical methods are employed in tutorial systems to compare student responses with a domain-model built from pre-constructed gold-standard answers (Graesser, Wiemer-Hastings, Wiemer-Hastings, Harter, & Person, 2000). In our context, such a static domain-modeling solution is impossible because of the wide quantitative and qualitative range of acceptable proofs, i.e., generally, our set of gold-standard answers is even infinite.

With regard to interpreting mathematical texts, (Zinn, 1999) and (Baur, 1999) present analyses of course-book proofs in terms of Discourse Representation Theory (Kamp & Reyle, 1993). The language in our dialogs is more informal: natural language and symbolic mathematical expressions are mixed more freely, there is a higher degree and more variety of verbalization, and mathematical objects are not properly introduced. Moreover, both above approaches rely on typesetting and additional information that identifies mathematical symbols, formulae, and proof steps, whereas our input does not contain any such information. Forcing the user to delimit formulae would reduce the flexibility of the system, make the interface harder to use, and would not guarantee a clean separation of the natural language and the non-linguistic content anyway.

Recent research into dialog modeling has delivered a variety of approaches more or less suitable for the tutorial dialog setting. For instance, scripting is employed in Autotutor (Person, Graesser, Harter, Mathews, & the Tutoring Research Group, 2000) and knowledge construction dialogs are implemented in Geometry Tutor (Matsuda & VanLehn, 2003). Static dialog scripts are not suitable in our case because they are in conflict with the requirement of flexibility for active learning. Outside the tutorial domain, the framework of Information State Update (ISU) has been developed in the EU projects TRINDI[1] and SIRIDUS[2] (Traum & Larsson, 2003), and applied in various projects targeting flexible dialog. An ISU-based approach with several layers of planning is used in the tutorial dialog system BEETLE (Zinn, Moore, Core, Varges, & Porayska-Pomsta, 2003).

Finally, the dialogs in our corpus reveal many challenges for human-oriented theorem proving. Traditional automated theorem provers (e.g., OTTER and Spass) work on a very fine-grained logic level. However, interactive proof assistants (e.g., PVS, Coq, NuPRL, Isabelle) and in particular proof planners (e.g., OMEGA and λClam) support abstract-level reasoning. The motivation for abstract-level reasoning is twofold: (a) to provide more adequate interaction support for the human and (b) to widen the spectrum of mechanizable mathematics. Proof assistants are usually built bottom-up from the selected base-calculus; this often imposes constraints on the abstract-level reasoning mechanisms and the user-interface.

3.3 Methods Applied

Due to the lack of suitable corpora to guide our research, we had to collect the required empirical data ourselves. To conduct the experiment, we implemented a Wizard-of-Oz support-tool that allows for the gradually developed components to be incorporated into the system simulation, thus refining the experimental setup. Dialog modeling in our project employs the Information State Update approach, and natural language processing is based on a combination of deep and shallow techniques. For mathematical domain reasoning, we employ and adapt techniques for abstract-level, human-oriented theorem proving and proof planning.

3.4 Results and Their Implications

The initial hypothesis in the DIALOG project has been that a dynamic combination and integration of (i) advanced solutions for natural language analysis (including deep semantic

[1] `http://www.ling.gu.se/research/projects/trindi/`
[2] `http://www.ling.gu.se/projekt/siridus/`

analysis) and generation, (ii) flexible tutorial dialog management, and (iii) cognitively adequate mathematical domain reasoning, is crucial for a tutorial natural language dialog on mathematical proofs. This hypothesis has been confirmed in the project. In particular, the phenomena we identified through corpus analysis demonstrate that deep semantic analysis is essential in our project context. They also motivate the use of a human-oriented, abstract-level approach to proof development. The explicit abstract-level representation of proof steps (logically sound or unsound) uttered by the students in our experiments is crucial for the analysis of their acceptability in a tutorial dialog context. Additionally, from a logical point of view, proofs steps are highly underspecified (e.g., logically relevant references are left implicit) causing an additional challenge for bridging the gap between input analysis and mathematical domain reasoning.

3.4.1 AP1: Experimental Test Environment and Empirical Investigations

In the first experiment, which has been prepared and carried out in cooperation with ERGOSIGN GmbH (Dieter Wallach), we collected a corpus that is highly interesting with respect to natural language aspects and mathematical domain reasoning. As a byproduct of our project we developed the DiaWoZ support tool for Wizard-of-Oz experiments. Investigation of the corpus resulted in an overwhelming list of key phenomena raising interesting and novel research challenges (see for instance (Benzmüller et al., 2003a; Benzmüller et al., 2003; Wolska & Kruijff-Korbayová, 2003; Autexier, Benzmüller, Fiedler, Horacek, & Vo, 2004)). This was not expected, in particular, because of the simplicity of the mathematical domain (naive set theory) chosen for this first experiment. Hence, we decided to first react on these results before conducting further experiments as was initially planned. Many of the identified phenomena are relevant not only for the tutorial natural language dialog context but have a much wider impact for natural language interactions in human-oriented theorem proving.

After describing **(A)** the DiaWoZ support-tool and **(B)** the experimental set-up, we **(C)** present examples from our corpus. Then we will discuss the identified phenomena from **(D)** the linguistic perspective, **(E)** the tutorial perspective, and **(F)** the mathematical domain reasoning perspective.

(A) The DiaWoZ Tool

For carrying out the experiment, we have implemented a tool, DiaWoZ, to support the wizard in his actions and to collect data on-line. DiaWoZ consists of two parts which are responsible for dialog specification (authoring) and dialog execution, respectively. Dialog authoring is more powerful than in standard approaches, since it combines a finite state machine with information states. Information states are conceived of as sets of local and global variables. Moreover, transitions have preconditions and effects, in terms of information states. Dialog execution follows these specifications, while the wizard performs the following actions: he interprets the user's utterance and generates the system output. The tool allows for the integration of gradually developed software components and supports progressive refinements of the modeled system architecture. The DiaWoZ tool is presented in (Fiedler & Gabsdil, 2002).

(B) The Experiment

The main goal of our experiment was to obtain a corpus of tutorial dialogs. A minor aim was to test the suitability of a previously defined algorithm implementing a socratic tutoring strategy.

24 subjects with varying background in humanities and sciences participated in the experiment. Their prior mathematical knowledge ranged from little to fair.

The experiment consisted of the following phases (with a fixed maximum duration):

(i) *Preparation and pre-test*: The subjects had to fill out a background questionnaire, to read lesson material on naive set theory, and to prove one theorem on paper.

(ii) *Tutoring session*: The subjects interacted with the tutoring system simulated by DiaWoZ. They had to prove the following theorems:

- $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$
- $A \cap B \in P((A \cup C) \cap (B \cup C))$
- Wenn $A \subseteq K(B)$, dann $B \subseteq K(A)$

(iii) *Post-test*: The subjects had to prove one further theorem on paper and to fill out a feedback questionnaire.

In phase (ii) the tutor-wizard's task was to respond to the student's utterances following a given algorithm. The wizard first classified the appropriateness of the subject's utterance (i.e. its completeness and accuracy) with respect to a valid proof of the theorem at hand. Then, the wizard decided which dialog moves to make next and verbalized them. Both subjects and wizard typed their input on a keyboard and inserted mathematical symbols by clicking on buttons on the DiaWoZ user interface.

The tutor-wizard employed one of three tutorial strategies: minimal feedback, didactic, socratic. Depending on the tutoring strategy, the dialog move options (specified in an algorithm) included informing the subject about the appropriateness of the utterance, giving hints on how to proceed further, explaining a step under consideration, prompting for the next step, or entering into a clarification dialog. The experiment is discussed in more detail in (Benzmüller et al., 2003b).

(C) The Corpus

The corpus (see also (Wolska et al., 2004)) comprises 66 recorded dialogs containing on average 12 turns. It contains 1115 sentences in total, of which 393 are student sentences. We give an example dialog from the corpus below[3] (**T** and **S** mark tutor and student turns respectively):

T1: *Bitte zeigen Sie: Wenn* $A \subseteq K(B)$*, dann* $B \subseteq K(A)$*!*

S1: $U \backslash A = K(A)$

T2: *Das ist richtig! Und weiter?*

S2: *wenn ich das wüsste*

T3: *Zuerst müssen wir die wenn-dann-Beziehung betrachten.*

S3: *also ist* $A \not\subseteq B$

T4: *Nein, das bringt nichts. Wissen Sie, was Sie mit der wenn-dann-Beziehung anfangen müssen?*

S4: *nein*

T5: *Sie müssen die wenn-dann-Beziehung auflösen, indem Sie die Wahrheit der Voraussetzung annehmen.*

S5: *schon klar*

T6: *Dann tun Sie das bitte.*

S6: *also ist* $A \subseteq K(B)$

T7: *Richtig. Und nun ?*

S7: *dann ist* $A \subseteq U \backslash B$

T8: *Ja, das stimmt. Was dann?*

S8: *also ist* $A \cap B = \emptyset$

T9: *Sehr gut.*

S9: *danke. also gilt ferner, da A und B keine gemeinsamen Elemente haben , dass K(A), definiert als* $U \backslash A$*, die Menge B enthält*

T10: *Können Sie das noch genauer erklären?*

[3] The corpus is available upon request by email to dialog@ags.uni-sb.de.

S10: *also: vorraussetzung ist: $A \subseteq K(B)$ daraus folgt: $A \cap B = \emptyset$ Wenn aber $A \cap B = \emptyset$, bedeutet das, A und B sind vollkommen verschieden.*

T11: *Ich meinte, daß Sie den letzten Schritt: also gilt ferner, da A und B keine gemeinsamen Elemente haben, dass K(A), definiert als $U \setminus A$, die Menge B enthält noch genauer erklären sollten.*

S11: *wie genauer*

T12: *Sie haben Ihre Schlußfolgerungen nicht genau ausgeführt, so daß ich sie nicht nachvollziehen kann.*

S12: *K(A) ist $U \setminus A$, und da A und B verschieden sind, muss $B \in K(A)$ sein*

T13: *Meinten Sie wirklich: $B \in K(A)$?*

S13: *nein .... $B \subseteq K(A)$*

T14: *Gut. Wie geht es dann weiter?*

S14: *das wars: wenn $A \subseteq K(B)$, dann sind A und B verschieden, haben keine gemeinsamen Elemente, daraus folgt, dass $B \subseteq K(A)$ sein muss*

T15: *Richtig. Damit ist der Beweis abgeschlossen. Ich wiederhole noch einmal: Aus $A \subseteq K(B)$ folgt $A \cap B = \emptyset$, denn gäbe es ein Element $x \in A \cap B$, dann wäre $x \in A$ und $x \notin K(B)$, was ein Widerspruch zur Annahme ist. Da $A \cap B = \emptyset$ gilt, muß also B komplett in K(A) enthalten sein, also $B \subseteq K(A)$, was den Beweis abschließt .*

To achieve a systematical investigation of the correlations between (i) domain-specific content and its linguistic realization, and (ii) the use, distribution, and linguistic realization of dialog moves in the collected data, we are annotating the corpus with dependency-based semantic relations that build up the linguistic meaning of the utterances (Wolska & Kruijff-Korbayová, 2004b), and with dialog moves (Tsovaltzi & Karagjosova, 2004).

For the annotation of dependency relations we adapted the set of *tectogrammatical relations* (TRs) used in the Prague Dependency Treebank (Hajičová, Panevová, & Sgall, 2000). Using TRs rather than surface grammatical roles provides a generalized view of the correlations between domain-specific content and its linguistic realization.

For the dialog move annotation, we are adopting a commonly accepted high-level standard taxonomy of DAMSL dialog moves (Allen & Core, 1997), which we extended for our specific domain of tutoring mathematical problem solving. A preliminary version of our annotation scheme in presented in (Tsovaltzi & Karagjosova, 2004).[4] To the DAMSL taxonomy we added a new dimension that specifies an utterance's task-level function. The *task dimension* captures functions that are particular to the task at hand and its manipulation, and hence to the genre. There are two sub-dimensions, namely proof task and tutoring task. At the tutoring task level, we annotate information pertaining to the proof-steps (for student utterances) and to the teaching method (for tutor utterances).

[4] This scheme is being tested.

The annotations are performed with the MMAX annotation tool (Müller & Strube, 2003). MMAX allows for annotating utterances at multiple levels, which facilitates finding out the above correlations.

To be able to test the reliability of the annotation schemes, we asked two annotators to perform annotations independently. We will have completed the annotation of dialog moves by the time of the review, and the annotation of semantic relations by the end of this project period.

(D) Phenomena from Linguistic Perspective

An analysis of the language phenomena observed in the corpus is presented in (Benzmüller et al., 2003; Wolska & Kruijff-Korbayová, 2003). They include: (i) varying degree of verbalizing formal content, (ii) tight interleaving of natural language with formulae (e.g., "B enthaelt **kein x** $\in A$"), (iii) reference to sub-formulae (e.g., "Dann gilt fuer **die linke Seite**, wenn $C \cup (A \cap B) = (A \cup C) \cap (B \cup C)$"), (iv) ambiguity and imprecision (e.g. "$(A \cap B)$ muss **in** $P((A \cup C) \cap (B \cup C))$ **sein**"), (v) informal reference to mathematical concepts (e.g. "B **vollstaendig ausserhalb von** A **liegen** muss"), (vi) informal description of proof steps (e.g. "Ich **zerlege** jetzt die Potenzmenge: $P(C \cup (A \cap B)) \supseteq P(C) \cup P(A \cap B)$") and (vii) ambiguity between relations, e.g., Cause, Result and Condition.

(E) Phenomena from the Tutorial Perspective

As part of the first experiment design, the tutor-wizard was given an algorithm for hinting (on paper), constraining his hinting moves during the socratic tutoring sessions. Other dialog moves were less restricted. The experiment served to test the coverage and suitability of the algorithm.

The pre- and post-test were used to evaluate the suitability of the different tutoring strategies (minimal feedback, didactic, socratic). According to expectations, subjects exposed to minimal feedback learned the least. Against expectations, though, the subjects of the didactic group learned more than the subjects of the socratic group. However, these results cannot be regarded as conclusive. One reason is that the students were exposed to the socratic tutoring method only for a short time while socratic tutoring supposedly deploys its efectiveness over a longer time span. Also uncertainties about the evaluation of the experimental data are raised by the provisional character of our categorization scheme and the tutor-wizard's difficulties when judging about the appropriateness of the student's utterance.

Tutorial aspects of the corpus study are discussed in more detail in (Fiedler & Tsovaltzi, 2003a; Tsovaltzi & Fiedler, 2003a).

(F) Phenomena from the Mathematical Domain Reasoning Perspective

Notion of Proof. For analyzing the notion of human-oriented mathematical proofs, mainly shaped-up textbook proofs have been investigated in the deduction systems community. Our corpus provides an important alternative view on it, since textbook proofs neither reveal the actual dynamics of proof construction nor do they show the weaknesses and inaccuracies of the user's utterances, i.e., the user's proof step directives. Our corpus also illustrates the style and logical granularity of human-constructed proofs. The style was mainly declarative, e.g, the students declaratively described the conclusions and some (or none) of the premises of their inferences. This is in contrast to the procedural style employed in most state-of-the-art proof assistants where proof steps are mainly invoked by calling rules/tactics/methods, i.e., some proof refinement procedures.

Our hypothesis that assertion level reasoning (Huang, 1994) plays an essential role has been confirmed. The phenomenon that assertion level reasoning may by highly underspecified in human-constructed proofs, however, is a novel result (Autexier et al., 2004).

Our insights stimulated the *interactive proof island* approach in the ΩMEGA project as addressed in (Siekmann et al., 2003) and the proof languages LU and LS discussed in Section 3.4.3; see also (Autexier et al., 2004) and (Hübner, Autexier, Benzmüller, & Meier, 2004).

Underspecification carries over to Mathematical Domain Reasoning. Underspecification is a well known phenomenon in natural language analysis. Our corpus reveals that underspecification also occurs in the content and precision of mathematical utterances (proof step specification) and thus carries over to domain reasoning (Autexier et al., 2004). Interestingly underspecification also occurs in shaped-up textbook proofs but has only very recently been addressed (Zinn, 2003).

To illustrate the underspecification aspect we briefly discuss an example from our corpus. The user utterance

U1: also folgt letztendlich : $K ((A \cup B) \cap (C \cup D)) = (K (A) \cap K (B)) \cup (K (C) \cap K (D))$.'

is logically strongly underspecified. Here, it is neither mentioned from what assertion(s) in the discourse this statement exactly follows nor how these assertions are used. However, such detailed information is typically required in proof assistants to execute the user's proof step directive, i.e., to 'understand' and 'follow' the user's argumentation.

3.4.2 AP2: Dialog Modeling

We employ the Information-State Update approach to dialog management developed in the TRINDI[5] and SIRIDUS[6] projects (Traum & Larsson, 2003). The Dialog Manager (DM) is the central component that connects and orchestrates other modules in our system. Our assumption so far in the project has been that a simple sequential data-flow between modules is sufficient. These include the analysis and generation modules, the tutorial component (including a hinting algorithm) and the proof manager, i.e., the interface to the domain reasoning tools.

The user's input is captured by the DM and passed on to the input interpretation module. The task of input interpretation is two-fold. Firstly, it is to construct a representation of the utterance's linguistic meaning. Secondly, it is to identify and separate within the utterance

1. parts which convey domain knowledge that should be verified by a domain reasoner; for example, the entire utterance "$\overline{(A \cup B)}$ ist laut deMorgan-1 $\overline{A} \cap \overline{B}$" can be evaluated, and
2. parts which constitute meta-communication with the tutor (e.g., "Ich habe die Aufgabenstellung nicht verstanden.") that are not to be processed by the domain reasoner.

The representation of the analyzed input is returned to the DM and merged with the discourse context maintained in the Information State. If there is no proof-relevant content, the corresponding meta-communication dialog moves trigger directly. In the presence of proof-relevant content, the DM passes it to the Proof Manager (PM) for evaluation by the theorem prover (Omega). The PM returns a proof step category of the input such as "correct", "complete-partially-accurate", "complete-inaccurate", "incomplete-accurate", "incomplete-partially-accurate", or "wrong" to the DM (Fiedler & Tsovaltzi, 2003a).

Correct input leads to straightforward acceptance. The system response in other cases depends on the employed tutorial strategy, namely, whether and what kind of hint the system produces. The hinting algorithm relies on the previous resolution of underspecified parts, which is one of the tasks of the proof manager. When a dialog move has been selected, the DM calls the

[5] `http://www.ling.gu.se/research/projects/trindi/`
[6] `http://www.ling.gu.se/projekt/siridus/`

generation module to produce the corresponding output realization.

In addition to modeling and implementation of the dialog manager, we also identified several continuative problems:

(A) Interleaving of Analysis, Tutoring, Domain Reasoning

In addition to the interleaving of tutoring and domain reasoning aspects in analyzing the appropriateness of proof steps we came across cases where domain reasoning is required during natural language analysis. Thus, the simple sequential process model sketched above is probably too weak for our purposes. This aspect of the link between analysis and domain reasoning will be investigated in more depth in the next project period.

(B) Accommodating ill-formed Input

When analysis fails, the system can initiate a correction subdialog. On the other hand, it is not desirable to go into syntactic details and distract the student from the main tutoring goal. In order to identify suitable practical strategies for handling ill-formed input, further experimentation is needed.

(C) Tutorial Aspects

We have carried out an investigation of the role of marked informationally redundant utterances (IRUs) in tutorial dialogs in the tutorial genre (Karagjosova, 2003). IRUs are utterances which do not provide new information in dialog, but refer to information that is supposed to be shared between speaker and hearer. The study suggests that marked IRUs make non-salient propositions salient, or maintain the salience of propositions in dialog. This is interpreted as an indication that marked IRUs may assist the active learning process since they aid the students in recognizing old information that may help activate related knowledge.

3.4.3 AP3: Architecture and Interfaces

The coarse grained set-up of the math tutoring environment is displayed in Figure 1 and discussed in more detail in (Benzmüller et al., 2003a). This model, enhanced by a more fine-grained specification of individual system components, such as the hinting algorithm, has been

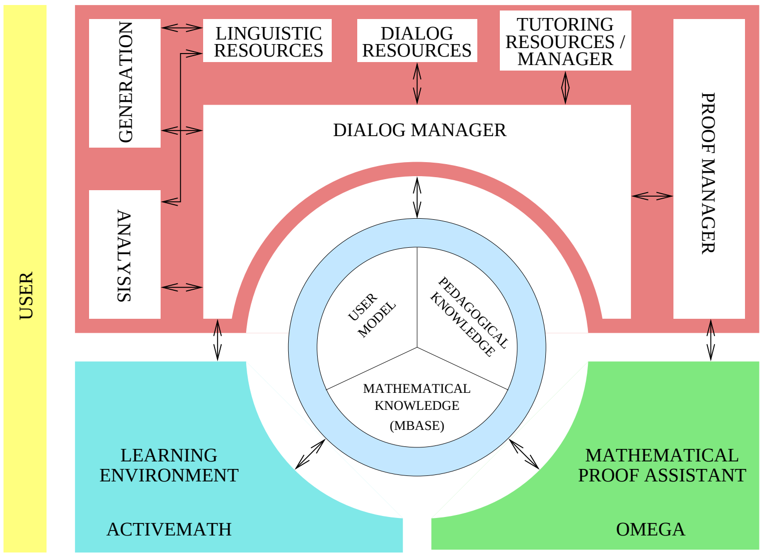

Figure 1. The math tutoring set-up.

guiding our project in the first phase. In particular, it provided frame constraints for the experimental set-up. Full realization and implementation of a system as sketched here requires an effort comparable, for example, to building a mathematical assistance environment such as ΩMEGA; typically such systems are developed by small research teams over more than a decade. The multi-disciplinarity and novelty of the task furthermore causes additional challenges. The focus of work in the current project phase has been on: (A) the interface between NL analysis and proof management, (B) the refinement of the hinting algorithm and its interplay with other modules, and (C) requirements and solutions for the collaboration between the proof manager and the domain reasoning tools provided by the ΩMEGA environment. The design of appropriate interfaces between the different modules is non-trivial. We therefore first concentrated on important individual challenges and delayed the top-down fine-graining of the joint overall architecture.

(A) Interface between Analysis and Proof Management

An important issue in the project was to analyze the notion of proof in our corpus. This analysis was guided by the question “Which are the typical, linguistically categorizable proof steps the students performed?” This lead to the following categories: (i) derivation of new facts, (ii) introduction of a new subgoal by referring to a replaced goal, (iii) decomposition of complex goal formulae by introduction of new hypotheses, (iv) assignment of values to instantiable variables, (v) introduction of abbreviations for complex formulae or subterms, and (vi) statement that the proof is finished.

Underspecification plays an essential role for all of these categories, for instance, in (i) we have potentially under-specified references to the used facts. Since linguistic analysis is not always able to categorize a given utterance uniquely, we introduce a non-deterministic branching over possible proofs (`Or`) to represent the different alternative interpretations. Finally, to obtain a richer proof representation format suitable for more complex proofs, we extend these identified steps in two ways: Firstly, for the goal reduction we allow for more than one subgoal. Secondly, we add case distinction. While these aspects do not occur in our corpus, it is clear that they generally play a role in mathematics.

The resulting proof language with underspecification, LU, is presented in (Autexier et al., 2004). LU stands for proof language with underspecification (or user-oriented proof language). LU is employed to represent the domain specific results of natural language analysis, i.e, the analysis module passes a (potentially) underspecified partial proof, represented in LU, to the dialog manager module where it is stored as part of the information state maintained by the dialog manager. The utterance U1 from above is for instance encoded in LU as:

Fact N_x **:** K ((A ∪ B) ∩ (C ∪ D)) = (K (A) ∩ K (B)) ∪ (K (C) ∩ K (D)) **from** $\bullet$

where N_x is an (automatically generated) name for the introduced fact and “$\bullet$” denotes underspecification.

(B) Refinement of the Hinting Algorithm and its Interplay with other Modules

An emphasis in the project has been on the refinement of the initial version of the hinting algorithm (Fiedler & Tsovaltzi, 2003b; Tsovaltzi, Horacek, & Fiedler, 2004). This work is guided by the phenomena we identified in the corpus. The hinting algorithm is part of the tutoring manager in Figure 1. Via the dialog manager it can access, for instance, the current partial proof as represented in the above language and the discourse structure. It furthermore accesses and updates additional information on the *Hinting Session Status* maintained by the tutoring manager. The session status, for instance, provides knowledge such as domain concepts used (defined in the enhanced ontology; see Section 3.4.4), proof steps and dialog turns performed, and the proof step category; see (Tsovaltzi & Fiedler, 2003a; Tsovaltzi & Karagjosova, 2004; Tsovaltzi et al., 2004) for details.

(C) Proof Management and Domain Reasoning

The tasks of the proof manager are many-fold. One prominent task is to employ domain reasoning in order to resolve underspecified parts in the LU representation of the proof. Further tasks are related to the student answer categorization as required by the hinting algorithm. Here, we briefly discuss some of the architecture and interface issues involved.

The proof manager maintains a representation of the proof development in order to address proof step evaluation requests and to provide a bridge to the domain reasoning tools. For this we developed the language LS which stands for system-oriented proof language. As requirements for LS we have identified:

1. Like LU, LS should support abstract-level representation of proof steps (since analysis of proof step granularity aspects is one of the proof managers tasks).
2. Unlike in LU proof steps should be fully specified in LS (thus, underspecification has to be resolved when translating between LU and LS).
3. Like LU and unlike the proof languages of most proof assistants, LS should support the representation of sound *and* unsound proof steps (since the correctness analysis is one of the proof managers tasks).
4. LS should also serve as an interface to various specialized domain reasoning tools (since we want to keep their concrete choice open).

LS is based on the notion of tasks developed in the OMEGA project; it is presented in more detail in (Hübner et al., 2004). The LS interface provides a novel top-down viewpoint on the development of interfaces for proof assistants, which is therefore relevant not only for the DIALOG project but more generally for proof assistants research.

An important task of the proof manager is *proof step evaluation*. The task is to analyze a user proof step with respect to the criteria *soundness*[7], *granularity* ('logical size' or 'argumentative size') of the step, and *relevance* (is the proof step needed or useful in order to achieve the proof goal?).

A further task of the proof manager is to support the resolution of underspecified knowledge within the translation from LU to LS. Our solution is to employ an abstract-level proof system that employs a heuristically guided approach to enumerate assertion-level proof step alternatives (Vo, 2003, To appear; Vo, Benzmüller, & Autexier, 2003a, 2003b, 2003). From these

[7] In some of our papers we call this also accuracy.

alternatives we then read off information that instantiates the underspecified data.

The translation between LU and LS, the resolution of underspecified parts, and the communication between the tutor module and the proof manager are still under development. For the demonstrator, we are currently working only with the LS language enriched with underspecification features so that it can also serve as an LU-alternative.

3.4.4 AP4: Provision and Preparation of Mathematical Knowledge

In this Section we report on the following activities: (A) modeling and enhancement of static mathematical knowledge in a mathematical knowledge base, (B) provision of dynamic mathematical knowledge through assertion level domain reasoning and knowledge retrieval, and (C) specific domain reasoning challenges.

(A) Modeling and Enhancement of Static Mathematical Knowledge

The ΩMEGA environment collaborates with the mathematical knowledge repository MBASE (Franke & Kohlhase, 2000). Besides other knowledge, MBASE maintains the formerly hardwired mathematical knowledge base of the ΩMEGA environment. Knowledge representation in MBASE and communication with MBASE-clients is supported by the OMDOC standard (Kohlhase, 2003) for open mathematical documents.

We encoded the theory corresponding to the lesson material on *naive set theory* prepared for our experiment, in OMDOC and added it to MBASE.

Due to their ontological structure, mathematical knowledge bases such as MBASE naturally provide some relations that are relevant for tutoring. An example is:

Hypotaxis: Mathematical concept σ is in hypotaxis to σ' if and only if σ' is defined using σ. We say, σ is a hypotaxon of σ'.
Examples: $\mathsf{hypotaxon}(\subseteq, Powerset)$, $\mathsf{hypotaxon}(\in, \cap)$

Direct hypotactic dependencies between concept definitions are often explicitly represented in mathematical knowledge bases while their transitive closure is usually left implicit provided that it can be efficiently computed from the given information.

Other interesting relations are *Antithesis* (e.g., $\mathsf{antithesis}(\in, \notin)$ and $\mathsf{antithesis}(\subset, \not\subset)$) and *Duality*. They are often not directly supported in mathematical knowledge bases; this has been also the case for MBASE.

We have therefore identified a variety of such relations that can effectively support hinting in our context. In addition to just relating concepts, we also consider relations between inference rules and other inference rules, and between inference rules and formulae (since inference rules are often procedurally encoded counterparts of declarative formulae) (Tsovaltzi & Fiedler, 2003b).

Currently, we are investigating whether all of these relations should be explicitly added to MBASE or whether we can provide algorithms that can efficiently analyze the knowledge base and automatically generate them.

(B) Assertion Level Reasoning and Knowledge Mediation

The two following hypotheses have been confirmed by our experiments: (i) assertion level reasoning plays an important role in proofs constructed by humans (in our domain) and (ii) support for the representation and direct processing of abstract-level proofs is needed in a tutorial context.

The proof language LS addresses these requirements and additionally serves as an interface to specific mathematical domain reasoners. A range of different mathematical reasoning systems is available in the OMEGA mathematical assistance environment.

ΩMEGA's proof planner MULTI directly supports abstract-level reasoning. It is currently adapted such that it directly operates on LS. Our novel top-down viewpoint helps to reduce the old planner's strong dependency on OMEGA's logical base layer (a higher-order natural deduction calculus).

Traditional automated theorem provers operate on fine-grained logic level; their relevance for our project is improved by mechanisms such as TRAMP (Meier, 2000) which translate fine-grained logic proofs into abstract-level human-oriented proofs. However, TRAMP has not yet been adapted to support LS proofs.

The role of our language LS thus is to provide a uniform and adequate interface to the heterogeneous domain reasoning tools, which may employ different logical calculi and operate on different levels of logical granularity.

The novel proof management tasks in our project require the improvement of direct assertion level reasoning since the respective mechanisms provided in proof planning or traditional automated theorem proving are too weak. As we have mentioned we are particularly interested in (resource-bound) enumerations of assertion-level proof step alternatives. We therefore developed and implemented in the project an agent-based assertion mediator. This tool mediates information between the LS proof and the static mathematical knowledge repository MBASE where, for instance, the assertions (i.e., axioms, definitions, and lemmata) of the naive set theory domain are stored. The mediator exploits the OANTS anytime-approach (Benzmüller & Sorge, 1999; Benzmüller, Jamnik, Kerber, & Sorge, 2001) to analyze the LS proof situation at hand and to search for applicable assertions in the knowledge repository. Each identified assertion is then dynamically suggested to the proof manager together with particular information on how it can be applied. Dynamic ranking of the set of applicable assertions is realized based on heuristic criteria. Further details on this assertion mediator are reported in (Vo, 2003, To appear; Vo et al., 2003a, 2003b, 2003).

3.4.5 AP5: Natural Language Interfaces

Input Understanding

Given the complexity of the language phenomena observed in the corpus, we adopted a methodology of deep analysis of the input text. Our objective has been to develop an approach to analyzing informal mathematical text in which (i) phenomena related to the tight interleaving of natural and mathematical languages would be accounted for and (ii) different degrees of the mathematical content verbalization would be treated uniformly. On the other hand, given the telegraphic nature of the language and common ungrammaticality, we also started to investigate how to combine the deep semantically oriented analysis with shallow techniques.

Our approach to analyzing the mixed natural language and mathematical language input is presented in (Wolska & Kruijff-Korbayová, 2003; Wolska & Kruijff-Korbayová, 2004a). We employ a dependency-based framework as a representation of the *linguistic meaning* (LM) of an utterance at a deep structure level. Our approach inherits from the Prague School dependency approach of Functional Generative Description (Sgall, Hajičová, & Panevová, 1986; Kruijff, 2001) where the central frame unit of a sentence/clause is the head verb which specifies the *tectogrammatical relations* of its dependents (*participants*). We use the grammar formalism of Multi-Modal Combinatory Categorial Grammar, where semantic dependency relations are explicitly encoded in the lexicon of the grammar as modal relations (Baldridge, 2002; Baldridge & Kruijff, 2003). The LM is represented using *Hybrid Logic Dependency Semantics* (HLDS) (Kruijff, 2001; Baldridge & Kruijff, 2002).[8] To achieve uniform analysis of inputs with different degree of mathematical content verbalization, the syntactic categories assigned in the lexicon to the mathematical expressions are treated in the same way as those of linguistic entries.

The analysis proceeds in three stages: (i) At the pre-processing stage, mathematical expressions are identified, analyzed, categorized, and substituted with default lexicon entries encoded in the grammar (e.g., CONSTANT, TERM, FORMULA, 0_FORMULA (formula missing left argument), etc.); (ii) Next, the input is syntactically parsed using openCCG[9], an open source MMCCG parser, and the LM representation is constructed compositionally along with the parse; (iii) The LM representation is subsequently embedded within an HLDS representation of the discourse context and (iv) interpreted by consulting a domain-specific ontology and a semantic lexicon of the domain. (v) Finally, the interpreted representation is translated into a LU representation. When domain interpretation is not unique (e.g. multiple domain concepts from the semantic lexicon or the ontology can be assigned to the literal meaning), the task of evaluation and disambiguation is left to the proof manager.

We illustrate the interpretation on an example from the corpus:

Nach deMorgan-Regel-2 ist $K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$

"ist" represents the meaning **hold**, and in this frame takes dependents in the tectogrammatical relations Criterion ("deMorgan-Regel-2") and Patient (the mathematical expression categorized as of type FORMULA). The semantics of this utterance is represented by the following HLDS:

@$h1$(holds $\wedge$ <CRITERION>($d1$ $\wedge$ deMorgan-Regel-2) $\wedge$ <PATIENT>($f1$ $\wedge$ FORMULA))

where $h1$ is the state where the proposition **hold** is true, and the nominals $d1$ and $f1$ represent the dependents of kinds Criterion and Patient respectively, of the head **hold**. The predicate **hold** and the discourse referents in the HLDS term are subsequently checked against an ontology and lexical semantics rules to find that **hold** represents an *assertion*, and "deMorgan Regel 2" is a lemma in naive set theory. The utterance is then translated into an underspecified proof-step representation:

[8]We have been collaborating with the NEGRA in SFB378 project in which the same approach is employed at the level of syntactic representation.

[9]`http://openccg.sourceforge.net`

Fact N_x **:** $K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$ **from** $\bullet$ **by** deMorgan-Rule-2

Our current work concentrates on applying the HLDS-based approach to discourse representation presented in (Kruijff & Kruijff-Korbayová, 2001).

Natural Language Generation

We use the proof presentation system *P.rex* (Fiedler, 2001) for the generation of the system's utterances. To account for our project setting *P.rex* had to be extended in various respects.

3.4.6 AP6: Realization of a Demonstrator

A demonstrator version of a math tutor system is under development and will be presented at the review meeting.

3.5 Comparison With Research Outside of the Collaborative Research Center

With the DIALOG project we have entered a novel and multi-disciplinary research field and our initial research results have proved relevant for different research communities. This is illustrated not least by the range of scientific publications or invited talks of the DIALOG group members. Our results on abstract level proof representation languages have furthermore stimulated new collaboration outside the SFB, for instance, a recently initiated collaboration with Prof. Fairouz Kamareddine at Heriot Watt University in Edinburgh, as well as stimulated research on this topic by the Mathematical Knowlegde Management community. Suitable abstract level representation languages for mathematics and respective support tools are key ingredients for the envisioned transition from pen-and-paper based mathematics to modern computer-supported environments.

Some of the concepts and research questions resulting from our DIALOG research have recently been taken up by the EU projects TALK (Talk & Look: Tools for Ambient Linguistic Knowledge)[10] and LeActiveMath[11] (Language-Enhanced, User Adaptive, Interactive eLearning for Mathematics).

[10] `www.talk-project.org`
[11] `www.leactivemath.org`

3.6 Open Issues

The purpose of the first project phase was to explore the challenges involved in using flexible dialog in tutoring mathematics, to identify the research questions specific to our system setup, and to start addressing them. The data collected in the first experiment resulted in identifying many more challenges than we had expected. This gave rise to a broad range of research questions pertaining to all components in the developed system. For this reason, we have postponed additional experiments, in which we planned to explore flexible multimodal tutorial dialogs in mathematics, including spoken language as well as graphical input and output. We will carry out a second experiment in the current project phase, to collect dialogs with spoken language and gesture interaction consisting of pointing by the mouse.

Of the research questions that we identified and started addressing in the first phase, we choose the most pressing and important ones to focus on in the next phase.

Bibliography

Allen, J., & Core, M. (1997). Draft of DAMSL: Dialogue act markup in several layers. *DRI: Discourse Research Initiative, University of Pennsylvania*.

Autexier, S., Benzmüller, C., Fiedler, A., Horacek, H., & Vo, B. Q. (2004). Assertion-level proof representation with under-specification. *Electronic Notes in Theoretical Computer Science*. (In print)

Baldridge, J. M. (2002). *Lexically specifi ed derivational control in combinatory categorial grammar.* Unpublished doctoral dissertation, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh, Edinburgh, UK. (`http://www.iccs.informatics.ed.ac.uk/~jmb/dissertation/index.%html`)

Baldridge, J. M., & Kruijff, G.-J. M. (2002). Coupling CCG with hybrid logic dependency semantics. In M. Diab & P. Resnik (Eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).* Philadelphia PA.

Baldridge, J. M., & Kruijff, G.-J. M. (2003). Multi-modal combinatory categorial grammar. In *Proceedings of the 10th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL'03).* Budapest, Hungary.

Baur, J. (1999). *Syntax und semantik mathematischer texte.* Diplomarbeit, Fachrichtung Computerlinguistik, Universität des Saarlandes, Saarbrücken, Germany.

Benzmüller, C., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Pinkal, M., Siekmann, J., Tsovaltzi, D., Vo, B. Q., & Wolska, M. (2003a). Tutorial dialogs on mathematical proofs. In P. Baumgartner, P. A. Cairns, M. Kohlhase, & E. Melis (Eds.), *Proceedings of the IJCAI 03 Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems* (pp. 12–22). Acapulco, Mexico.

Benzmüller, C., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Pinkal, M., Siekmann, J., Tsovaltzi, D., Vo, B. Q., & Wolska, M. (2003b). A wizard-of-oz experiment for tutorial dialogues in mathematics. In V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi,

H. Pain, F. Verdejo, & K. Yacef (Eds.), *AIED2003 — Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education* (Vol. VIII: Advanced Technologies for Mathematics Education, pp. 471–481). Sydney, Australia: School of Information Technologies, University of Sydney.

Benzmüller, C., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Tsovaltzi, D., Vo, B. Q., & Wolska, M. (2003). Language phenomena in tutorial dialogs on mathematical proofs. In I. Kruijff-Korbayová & C. Kosny (Eds.), *Proceedings of DiaBruck'03, the 7th Workshop on the Semantics and Prgmatics of Language* (pp. 165–166). Saarbrücken, Germany.

Benzmüller, C., Jamnik, M., Kerber, M., & Sorge, V. (2001). Experiments with an agent-oriented reasoning system. In F. Baader, G. Brewka, & T. Eiter (Eds.), *KI 2001: Advances in Artificial Intelligence, Joint German/Austrian Conference on AI, Vienna, Austria, September 19-21, 2001, Proceedings* (pp. 409–424). Springer.

Benzmüller, C., & Sorge, V. (1999). Critical Agents Supporting Interactive Theorem Proving. In *Proceedings of 9th Portuguese Conference on Artificial Intelligence (EPIA-99).* Springer.

Fiedler, A. (2001). Dialog-driven adaptation of explanations of proofs. In B. Nebel (Ed.), *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1295–1300). Seattle, WA: Morgan Kaufmann.

Fiedler, A., & Gabsdil, M. (2002). Supporting progressive refinement of Wizard-of-Oz experiments. In C. P. Rose & V. Aleven (Eds.), *Proceedings of the ITS 2002 — Workshop on Empirical Methods for Tutorial Dialogue Systems* (pp. 62–69). San Sebastián, Spain.

Fiedler, A., & Tsovaltzi, D. (2003a). Automating hinting in mathematical tutorial dialogue. In *10th Conference of the European Chapter of the Association for Computational Linguistics—Proceedings of the Workshop on Dialogue Systems: interaction, adaptation and styles of management* (pp. 45–52). Budapest, Hungary.

Fiedler, A., & Tsovaltzi, D. (2003b). Automating hinting in an intelligent tutorial system. In *Proceedings of the IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems* (pp. 23–35). Acapulco.

Franke, A., & Kohlhase, M. (2000). System description: MBASE, an open mathematical knowledge base. In D. McAllester (Ed.), *Proceedings of the 17th Conference on Automated Deduction.* Springer.

Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., & Person, N. (2000). Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments, 8.*

Hajičová, E., Panevová, J., & Sgall, P. (2000). *A manual for tectogrammatical tagging of the Prague Dependency Treebank* (Tech. Rep. No. TR-2000-09). Prague, Czech Republic: Charles University.

Huang, X. (1994). Reconstructing Proofs at the Assertion Level. In A. Bundy (Ed.), *Proceedings of the 12th Conference on Automated Deduction* (pp. 738–752). Springer.

Hübner, M., Autexier, S., Benzmüller, C., & Meier, A. (2004). Interactive theorem proving with tasks. *Electronic Notes in Theoretical Computer Science*. (To appear)

Kamp, H., & Reyle, U. (1993). *From discourse to logic.* Dordrecht, the Netherlands: Kluwer Academic Publishers.

Karagjosova, E. (2003). Marked informationally redundant utterances in tutorial dialogue. In I. Kruijff-Korbayová & C. Kosny (Eds.), *Proceedings of DiaBruck'03, the 7th Workshop on the Semantics and Prgmatics of Language.* Saarbrücken, Germany.

Kohlhase, M. (2003). OMDoc*: An open markup format for mathematical documents (version 1.2)* [Open Specification]. (To appear in Springer LNAI; Manuscript at `http://www.mathweb.org/src/mathweb/omdoc/doc/spec/omdoc.ps`)

Kruijff, G.-J., & Kruijff-Korbayová, I. (2001). A hybrid logis fomralization of information structure sensitive discourse interpretation. In V. Matoušek, P. Mautner, R. Moučcek, & K. Taušer (Eds.), *Proceedings of the 4th International Conference on Text, Speech and Dialogue (TSD'2001), Železná Ruda, Czech Republic* (pp. 31–38). Springer.

Kruijff, G.-J. M. (2001). *A categorial-modal logical architecture of informativity: Dependency grammar logic & information structure.* Ph.d. dissertation, Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Matsuda, N., & VanLehn, K. (2003). Modelling hinting strategies for geometry theorem proving. In *Proceedings of the 9th International Conference on User Modeling.* Pittsburgh, PA.

McAllester, D. (Ed.). (2000). *Proceedings of the 17th Conference on Automated Deduction* (No. 1831). Springer.

Meier, A. (2000). TRAMP: Transformation of Machine-Found Proofs into Natural Deduction Proofs at the Assertion Level. In D. McAllester (Ed.), *Proceedings of the 17th Conference on Automated Deduction.* Springer.

Moore, J. (1993). What makes human explanations effective? In *Proc. of the 15th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ.

Müller, C., & Strube, M. (2003). Multi-level annotation in MMAX. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue.* Sapporo, Japan. (http://www.eml.villa-bosch.de/english/Research/NLP/Publications)

Person, N. K., Graesser, A. C., Harter, D., Mathews, E., & the Tutoring Research Group. (2000). Dialog move generation and conversation management in AutoTutor. In C. P. Rosé & R. Freedman (Eds.), *Building Dialog Systems for Tutorial Applications—Papers from the AAAI Fall Symposium* (pp. 45–51). North Falmouth, MA: AAAI press.

Sgall, P., Hajičová, E., & Panevová, J. (1986). *The meaning of the sentence in its semantic and pragmatic aspects.* Dordrecht, The Netherlands: Reidel Publishing Company.

Siekmann, J., Benzmüller, C., Fiedler, A., Meier, A., Normann, I., & Pollet, M. (2003). Proof development in OMEGA: The irrationality of square root of 2. In F. Kamareddine (Ed.), *Thirty fi ve years of automating mathematics* (p. 271-314). Kluwer Academic Publishers. (ISBN 1-4020-1656-5)

Traum, D. R., & Larsson, S. (2003). The information state approach to dialogue management. In J. van Kuppevelt & R. Smith (Eds.), *Current and new directions in discourse and dialogue.* Kluwer. (`http://www.ict.usc.edu/~traum/Papers/traumlarsson.pdf`)

Tsovaltzi, D., & Fiedler, A. (2003a). An approach to facilitating reflection in a mathematics tutoring system. In V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo, & K. Yacef (Eds.), *AIED2003 — Supplementary Proceedings of the 11th International Conference on Artifi cial Intelligence in Education* (Vol. V: Learner Modelling for Reflection, pp. 278–287). Sidney, Australia: School of Information Technologies, University of Sydney.

Tsovaltzi, D., & Fiedler, A. (2003b). Enhancement and use of a mathematical ontology in a tutorial dialog system. In A. Jönsson, J. Alexandersson, T. Becker, K. Jokinen, & M. Merkel (Eds.), *Proceedings of the IJCAI 03 Workshop on Knowledge and Reasoning* *in Practical Dialogue Systems* (pp. 19–28). Acapulco, Mexico.

Tsovaltzi, D., Horacek, H., & Fiedler, A. (2004). Building hint specifications in a NL tutorial system for mathematics. In *Proceedings of the 16th International Florida AI Research Society Conference (FLAIRS-04).* Florida, USA. (In print)

Tsovaltzi, D., & Karagjosova, E. (2004). A dialogue move taxonomy for tutorial dialogues. In *Proceedings of 5th SIGdial Workshop on Discourse and Dialogue.* Boston, USA. (In print.)

Vo, B. Q., Benzmüller, C., & Autexier, S. (2003a). Assertion application in theorem proving and proof planning. In *Proceedings of the International Joint Conference on Artifi cial Intelligence (IJCAI).* Acapulco, Mexico. (poster description))

Vo, B. Q., Benzmüller, C., & Autexier, S. (2003b). Assertion application in theorem proving and proof planning. In *Proceedings of the 10th Workshop on Automated Reasoning: Bridging the Gap between Theory and Practice.* Liverpool, England.

Vo, Q. B. (2003). A task-oriented agent-based mechanism for theorem proving. In J. Liu, B. Faltings, M. Klusch, & C. Liu (Eds.), *The 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003)* (p. 275-281). IEEE Computer Society/WIC.

Vo, Q. B. (To appear). A task-oriented information mediator for mathematical assistant systems. *International Journal of Web Intelligence and Agent Systems*.

Vo, Q. B., Benzmüller, C., & Autexier, S. (2003). *An approach to assertion application via generalized resolution* (SEKI Report No. SR-03-01). Fachrichtung Informatik, Universität des Saarlandes, Saarbrücken, Germany.

Wolska, M., & Kruijff-Korbayová, I. (2003). Issues in the interpretation of input in mathematical dialogs. In *Proceedings of the Lorraine-Saarland Workshop Series on Prospects and Advances in the Syntax/Semantics Interface.* Nancy, France.

Wolska, M., & Kruijff-Korbayová, I. (2004a). Analysis of mixed natural and symbolic language input in mathematical dialogs. In *In Proceedings of the 42nd Meeting of the 42nd Annual Meeting of the Association for Computational Linguistics.* Barcelona, Spain. (Submitted)

Wolska, M., & Kruijff-Korbayová, I. (2004b). Building a dependency-based grammar for parsing informal mathematical discourse. In *the workshop Beyond Shallow Analyses of The 1st International Joint Conference on Natural Language Processing.* Sanya City, Hainan Island, China. (Accepted)

Wolska, M., Vo, B. Q., Tsovaltzi, D., Kruijff-Korbayová, I., Karagjosova, E., Horacek, H., Gabsdil, M., Fiedler, A., & Benzmüller, C. (2004). An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proceedings of LREC.* (In print)

Zinn, C. (1999). Understanding mathematical discourse. In *Proc. of Amstelogue'99.* Amsterdam.

Zinn, C. (2003). Computational framework for understanding mathematical discourse. *Logic Journal of the IGPL, 11*(4), 457–484.

Zinn, C., Moore, J. D., Core, M. G., Varges, S., & Porayska-Pomsta, K. (2003). The be&e tutorial learning environment (beetle). In *Proceedings of Diabruck, the 7th Workshop on the Semantics and Pragmatics of Dialogue.* Saarbrücken, Germany.