Project MI 4:

OMEGA:
Agent-oriented Proof Planning

3.1 General Information About the Project MI 4

3.1.1 Topic

Agent-oriented Proof Planning

3.1.2 Discipline and Field

The disciplines are computer science and artificial intelligence (AI).

3.1.3 Directors

Name, date of birth:	Prof. Dr (PhD) Siekmann, Jörg[1], 5.8.1941 Dr. Autexier, Serge[2], 02.10.1971 Dr. Benzmüller, Christoph[3], 8.9.1968	
Work address:	[1,3] FR 6.2 Informatik, Saarland University, P.O. Box 15 11 50, 66041 Saarbrücken, Germany [2] Deutsches Forschungszentrum für künstliche Intelligenz (DFKI GmbH), Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany	
Phone number:	[1] +49-681-302-4754 [2] +49-681-302-5073	[3] +49-681-302-4574
Fax number:	[1,3] +49-681-302-5076	[2] +49-681-302-2235
Email address:	[1] `siekmann@ags.uni-sb.de` [2] `autexier@dfki.de` [3] `chris@ags.uni-sb.de`	

Is the position of any of the project directors temporary?

Prof. Jörg Siekmann:	[X] No	[ ] Yes, ending on
Dr. Serge Autexier:	[ ] No	[X] Yes, ending on 31.08.2004
Dr. Christoph Benzmüller:	[ ] No	[X] Yes, ending on 31.12.2006

3.1.4 Transfer of the Project

The project is not being transferred to the Transfer Center.

3.1.5 Planned Studies

The following studies are planned as part of the project:

Studies with human subjects	[ ] Yes	[X] No
Studies with human embryonal stem cells	[ ] Yes	[X] No
Clinical studies on somatic cell or gene therapy	[ ] Yes	[X] No
Experiments with animals	[ ] Yes	[X] No
Studies involving genetic technology	[ ] Yes	[X] No

3.1.6 Previous and proposed funding of the project so far in the context of the collaborative research center (*external funding*)

Fiscal year	Personnel costs	Administrative expenses	Investments	Total
Through 2001	375.5			
2002	140.5			
2003	140.5			
2004	140.5			
Subtotal	797.0		24.0	
2005	166.8	4		
2006	166.8	4		
2007	166.8	4		

(Amounts in 1000s of EUR)

3.2 Summary

mehr high-level vision des systems, das man nun die enwtickelten techniken mit anderen kombineiren will, den reasoning level hochsetzen, um ein system zu bekommen, mit dem man dann math. Aktiv. unterstuetzen kann, wie z.b. publikationen und tutoring.

EDNOTE! Dies muss komplett neu geschrieben werden

The main objectives of the project OMEGA (MI4) are to further improve and better integrate reactive and deliberative proof planning and to extend the mathematical assistant by mechanisms in order to enable its integration into typical mathematical activities.

With respect to proof planning, the deliberative and reactive approaches provide complementary technologies for the encoding of mathematical problem solving techniques. On the basis of this technologically stable proof planning paradigms, the last phase of this project concentrates on the qualitative enhancement of proof planning by exchanging the underlying natural deduction calculus for the CoRe-calculus which supports proof development directly on the assertion level. This should overcome the limitations imposed by the cumbersome natural deduction calculus to the proof planning process. Furthermore, from a knowledge engineering point of view, it will result in a much leaner and mathematically more concise collection of proof planning knowledge, which again will increase proof planning efficiency, yield more concise proof plans, increase acceptance by humans and reduce the maintenance effort.

The possibility for proof development on the assertion level paired with more concise proof plans will foster a better integration of the mathematical assistant system directly into typical mathematical activities, such as writing publications. Furthermore, the goal to provide support for typical mathematical activities gives rise to research tasks concerned with the maintenance of mathematical knowledge, for instance formalized mathematical theories, domain specific proof planning methods as well as proofs and proof plans for specific problems. To this end starting from techniques developed for the management of large formal software developments we will investigate how these techniques can be transfered and extended to provide efficient management support for the aforementioned kinds of mathematical knowledge.

This results in a tool for mathematical knowledge management in a mathematical assistant, which should not only be a passive database of mathematical knowledge, but should incorporate mechanisms to pro-actively search and provide additional knowledge to the proof planning process. Thereby we investigate how the reactive suggestion mechanism developed in previous phases of the project could be adapted to design agents that actively extract available knowledge from the database which could contribute to the solution of an actual proof planning problem.

The developed technological solutions with respect to assertion level proof planning, management of mathematical knowledge and intelligent querying and integration of available mathematical knowledge will be evaluated by considering two typical mathematical activities. The first activity is the preparation of mathematical publications, and there especially the development of proofs in a style as can be found in mathematical textbooks. The second activity is tutoring mathematics students and will take place in cooperation with the SFB project DIALOG (MI3). Thereby, the mathematical assistant will be used as a formal backend to the proof manager developed in the DIALOG project in order to represent partial assertion level proofs resp. proof plans. Furthermore, it shall provide as far as possible feedback to the proof manager of the DIALOG-system about the relevancy by using proof planning technology.

3.3 State of the Art

The current state of the art, especially with respect to assertion level proof development, proof plannning, mathematical knowledge management, integration of mathematical knowledge into a reasoning process, and system support for typical mathematical activities has been partly shaped by the proposers. In the following sections we first give an overview of the state of the art and present in more detail the specific contributions of the proposers in Section 3.4. Furthermore, we refer to the SFB report for the project OMEGA (MI4).

Assertion-level Proof Development. Assertion-level proofs have long been recognised as a proof style that comes close to the style proofs are written by humans. The key inference rule is the application of assertions, which is a generic term to denote usable axioms, lemmas and hypotheses and they have been used as a basis for the generation of proof presentations in natural language (Huang, 1994a; Fiedler, 2001a). However, so far proofs are usually not directly constructed on the assertion level, but are obtained by transformation of an underlying proof, e.g. in natural deduction or resolution. The only exceptions are the focusing proofs for linear logic (Andreoli, 2000) and the assertion-level proof format for constructive first-order logic of (Abel, Chang, & Pfenning, 2001). The former derives new sequent calculus inference rules from assertions which can then be used during proof construction. However, it provides no support for the application of assertions to subformulas, as for instance the application of conditional equations. The latter builds up on the idea of focusing proofs to define proof checking rules for assertion-level proofs written by humans, e.g. by students, and provides no support to compute the actual assertion application.

Proof-Planning.

Proof planning was first introduced by Bundy in 1988. Bundy's key idea was to augment individual tactics with pre- and postconditions that specify the applicability of the tactic as well as its effects with respect to a proof state. This results in AI-planning operators, so-called methods. A proof planner searches for a plan, i.e., a sequence of methods, that derives the theorem from the given assumptions. The representation of a proof, at least while it is developed, consists of a sequence of abstract steps. The complete abstract proof (or parts of it) can be expanded to a logic-level proof. This enables automated proof search at an abstract level and a separated checking process.

Bundy and his group developed the first proof planner, CLAM (Bundy, Harmelen, Hesketh, & Smaill, 1991), in the early 1990s and applied it to prove theorems by mathematical induction. To guide the search of inductive proofs the rippling search heuristic for difference reduction (Hutter, 1990; Bundy, Harmelen, Ireland, & Smaill, 1990) is encoded into CLAM methods. Later on CLAM was re-implemented their the system λCLAM (Bundy, Harmelen, Horn, & Smaill, 1990; Richardson, Smaill, & Green, 1998). Thereby they introduced a tactical-like language to describe proof plan strategies and the expansion of complex methods. These techniques are now used in the ISAPLANNER (Dixon & Fleuriot, 2003) which is currently under development.

Another proof planner is part of the ΩMEGA system (Siekmann, Benzmüller, Fiedler, Meier, & Pollet, 2002). ΩMEGA is a proof development system for knowledge-based interactive and automated proof construction developed in our group since the mid 1990s (e.g., see (Huang et al., 1994; Benzmüller et al., 1997)). The development of ΩMEGA was motivated by the conviction that the solution of main-stream mathematical problems requires the combination of theorem proving based on mathematical knowledge with powerful reasoning experts such as machine-oriented theorem provers, computer algebra systems, or constraint solvers. ΩMEGA employs proof planning as the main tool for automated proof construction as a means to integrate mathematical knowledge and external expert systems into the theorem proving process.

One difference between proof planning in ΩMEGA and CLAM is the handling of heuristic control knowledge. Methods in CLAM (resp. λCLAM) include heuristic conditions about the desirability of the application of the method, while methods ΩMEGA include only the legal conditions. The heuristic conditions are explicitly encoded as control rules and are explicit parameters to the proof planner. Within the planner they guiding the search by reasoning on alternatives at choice points. Thereby control rules can encode both general and domain-specific mathematical control knowledge, since they can reason about the current proof planning state as well as the entire history of the proof planning process.

Mathematical Knowledge Management. This topic is of growing interest worldwide, which is documented by the many research initiatives that try to gradually increase the amount of computer-based support into mathematical practice. Prominent examples are EULER (`www.emis.de/projects/EULER/`), MKMNet (`monet.nag.co.uk/mkm/index.html`), MKM North America (`imps.mcmaster.ca/na-mkm-2004/`), Trial Solution (`www.trial-solution.de/`), LIMES (`www.emis.de/projects/LIMES/description.

`html`), ERAM (`www.emis.de/projects/JFM/`), OPENMATH and MOWGLI (`www.mowgli.cs.unibo.it/`), and the American QPQ (`www.qpq.org`) repository of deductive software tools.

Most theorem proving systems have their own database[1] or at least a collection of files containing formalized theories, but typically in the proprietary input format of the theorem proving system. The research about cooperation of theorem proving systems on the one hand side and the interest in databases for mathematical documents with different levels of formalizations on the other hand showed that databases for mathematical documents are necessary, which are not tailored to specific systems, are distributed over different places, and provide intelligent facilities to query the mathematical documents. This spurred the development of standard languages to encode mathematical knowledge, such as the OPENMATH-standard and its extension OMDOC and databases such as MBASE (Franke & Kohlhase, 2000; Franke, 2003). These databases, like the proprietary database of COQ, provide so far possibilities to add or change (parts of) mathematical documents and simple syntactical querying facilities, such as searching for symbols or pattern matching inside formulas.

Integration of Mathematical Knowledge. The mathematical knowledge that can be fruitfully integrated into a theorem proving process can be anything that may help the proof search process. This are axioms or lemmas not yet included into the lemma-base of the theorem prover, additional tactics, methods, control rules and strategies for a proof planner, as well as values computed by external computer algebra systems, models computed by model generators, or proofs of subgoals performed by an external theorem prover.

Little research has been devoted so far to the incremental integration of further axioms and lemmas that goes beyond simple querying in the database. More research has been devoted to the integration of external computer algebra systems or automatic theorem provers. For instance was the proof-planner λCLAM connected to the HOL interactive theorem prover (Gordon & Melham, 1993), the inductive theorem prover INKA was integrated into the tool for formal software development VSE, and an open platform to integrate CAV and CASE-tools was developed in the PROSPER project (Dennis et al., 2000).

The MATHWEB-software bus (Franke & Kohlhase, 1999; Zimmer & Kohlhase, 2002) was developed in our own group to provide a virtual software bus of mathematical tools, like for instance computer algebra systems, automated theorem provers, and model generators, which are distributed over the internet. The network is organized with brokers, on which clients can request specific mathematical services. Although so far a client has to know the input

[1] Even sometimes distributed, as for instance the Coq libraries(Team, 2003).

format of the requested tool the MATHWEB is already heavily used by various partners in Europe and the USA (Zimmer, Armando, & Giromini, 2001; Zimmer, Franke, Colton, & Sutcliffe, 2002; Benzmüller, Giromini, Nonnengart, & Zimmer, 2002; Zimmer & Dennis, 2002). However, it is currently under re-development to provide a problem oriented interface for the clients and include limited automatic reasoning capabilities into the brokers to find and possibly combine mathematical service to solve these stated problems (Zimmer, 2004). This work is executed in cooperation with the development of the MATHBROKER-tool (Caprotti & Schreiner, November, 2002), which, however, is more oriented towards computer algebra systems.

Support for Typical Mathematical Activities. There is a lot of activities towards integrating theorem proving tools into mathematical practice. One of the earliest projects in this perspective and with the objective to write formalized and yet readable mathematical documents was the Automath-Project (DeBruijn, 1970), which aimed at defining a mathematical vernacular and still is an active research topic (Kamareddine & Nederpelt, 2004). The Theorema project (Buchberger et al., 1997) has similar objectives, and develops extensions to the computer algebra system MATHEMATICA by domain specific theorem provers. It already provides some support for the preparation of mathematical documents with formal contents by exploiting the Mathematica notebook facilities.

Similar research has been conducted in the COQ community, by connecting a WYSIWIG-editor to the Coq proof assistant (Audebaud & Rideau, 2003) or providing graphical presentation tools to present proofs for geometrical problems performed in COQ (Bertot, Guilhot, & Pottier, 2003).

Aside from these activities to support the preparation of mathematical publications, research is devoted to use proof assistants for teaching mathematics. Thereby the proof assistant is used as part of an interactive learning environment not only to teach mathematics, but also to assist the student in solving exercises or proving mathematical properties. Examples are the SFB project DIALOG (MI 3), the Activemath project (Melis & Ullrich, 2001), (Abel et al., 2001), and the *Carnegie Proof Tutor (CPT)* (Scheines & Sieg, 1993).

3.4 Relevant Previous Work of the Proposers

Previous work of the proposers is discussed in more detail in the SFB report. Here we will only summarize the most relevant aspects; the focus will be on: assertion-level proof development, proof planning, mathematical knowledge management, integration of mathematical knowledge and support for mathematical activities.

Assertion-Level Proof Development. The notion of assertion-level proofs was coined in our group in the early nineties and use as an intermediate representation suitable for the natural language presentation of formal proofs (Huang, 1994b; Fiedler, 2001b, 2001a). However, direct proof construction on the assertion level was only possible by specialised tactics constructing a natural deduction derivation validating the assertion application. The support for direct assertion-level proof construction was enabled by the CORE-system (Autexier, 2003) within the last phase of the SFB. The CORE-system defines a communication infrastructure as a mediator between the user and the automatic reasoning procedures. It is based on the CORE-calculus, which is a uniform meta proof theory for contextual reasoning and encompasses most aspects of communication from the presentation of the proof state, via the supply of relevant contextual information about possible proof continuations, to the support for a hierarchical proof development. The proof theory is uniform for a variety of logics and exploits proof theoretic annotations in formulas to provide an assertion-level contextual reasoning style to the user and the automatic proof procedures. It is thus as far as possible intuitive for the user while at the same time still adequate for automatic reasoning procedures. Furthermore, the CORE-system provides the infrastructure required to accomodate both the use and the explicit representation of hierarchies that are inherent in problem solving in general.

Proof-Planning and Interactive Theorem Proving. The ΩMEGA-project is one of the leading groups in proof-planning research and developed the areas of knowledge-based proof planning. We first separately developed the deliberative proof-planner MULTI and the reactive, ressource-adaptive agent-based techniques for interactive theorem proving.

The MULTI-system is a general proof-planner parameterised over heuristic mathematical knowledge, like methods and control rules, as well as different strategies for checking the applicability of methods, selection of applicable methods, instantiation of meta-variables and backtracking. We developed several general and domain-specific strategies, which were successflly evaluated in different mathematical domains. Case studies were performed in the following proof-planning domains: limit problems (Melis & Siekmann, 1999; Meier, 2004), residue class problems (Meier, Pollet, & Sorge, 2001), problems of permutation groups (Cohen, Murray, Pollet, & Sorge, 2003), homomorphism problems (Meier, 2004), and problems about the irrationality of square roots of natural numbers (Meier, 2004).

In these case studies mathematical meta-reasoning was developed as control rules for MULTI. More specifically, meta-reasoning on failed proof attempts is used to guide backtracking or plan modifications, meta-reasoning about the content of the constraint store if it fails to compute instantiations for meta-variables, and meta-reasoning about goal dependencies due to shared meta-variables. Furthermore, the multi-strategy proof-planner MULTI was used to combine deliberative proof-planning and with the agent-based techniques for interactive theorem proving. The agent-based Ω-ANTS-mechanism was used as a special strategy of MULTI to reactively compute applicable methods. This allowed for a combination of reactive strategies to determine applciable methods with deliberative strategies such as method selection and backtracking.

Complementary to the multi-strategy proof-planner we developed the LEARNΩMATIC-tool to learn new methods and respective control knowledge. It creates new methods by generalizing sample proof plans to capture the contained proof patterns and also supports incremental learning of control knowledge. The LEARNΩMATIC-tool was evaluated by performing case studies in the domains of group theory, set theory and residue classes. As a result not only were the proof plans found using learnt methods shorter than those found without learnt methods, but even for some problems in group theory finding a proof plan was only possible using learnt of methods.

Mathematical Knowledge Management. The proposers developed the MAYA-tool (Hutter, 2000; Mossakowski, Autexier, & Hutter, 2001; Autexier, Hutter, Mossakowski, & Schairer, 2002; Autexier & Mossakowski, 2002) which was originally developed to support formal software development. The tool is based on the notion of development graphs, which allows for the structured representation of formal theories. Furthermore, MAYA incorporates incorporates a uniform mechanism for verification in-the-large to exploit the structure of the formalised theories, and maintains the proof work already done when changing the parts of the formalisation.

The notion of development graphs is also included in the definition of the OMDOC-standard to represent both informal and formal mathematical knowledge (Kohlhase, 2000). The OMDOC-language is an extension to the OPENMATH-standard by providing, between others, structuring of formal theories based on development graphs. The development of OMDOC started in our group, and still is developed in close cooperation with the members of the group.

Together with the conception of the OMDOC-language our group was and still is actively involved in the development of the MBASE-database (Franke & Kohlhase, 2000; Franke, 2003). The objective of MBASE is store mathematical knowledge, such as documents, formalized mathematical domains as well as exercises. Furthermore, MBASE provides basic facilities to query and extract the contained knowledge in OMDOC-format. Finally, it has been conceived to support the exchange of mathematical knowledge contained in different MBASEs distributed over different physical locations.

Integration of Mathematical Knowledge. The group has developed several techniques for integrating mathematical knowledge into theorem proving and proof-planning. First, members of the group initiated the the MATHWEB-software bus, whose development has grown into a separate research topic. The MATHWEB-SB provides mathematical services such as theorem provers, constraint solvers and computer algebra systems organized in a network of brokers. The brokers are situated at different physical locations and assist the clients to distribute mathematical problems to the integrated systems. Depending on the type of the called system and the submited problem, the computed solutions are proofs in some calculus, values for constraint variables, or computations. For each type of solution, several systems have been developed in the group to integrate them into the actual proof respectively proof plan. For proofs, we developed the TRAMP system that translates proofs in some calculus into an ΩMEGA-proof plan object. For computations, the SAPPER-system constructs a proof-plan from the trace of a computer algebra computation.

Inside the proof-planner or the interactive theorem prover, access to these systems is granted via specific tactics or proof-planning methods. In order to develop automatize proof development, the proof-planner uses the respective methods, and can even exploit the external systems inside control rules and strategies, which define the proof-plan search strategy. Similarly, agents have been developed for the external systems, which thereby reactively propose to the user solutions computed in parallel by external systems.

Aside from the integration of external reasoning systems, research was devoted to the integration of further knowledge from the mathematical database MBASE. Again the agent-based infrastructure has been successfully applied to develop a prototype of a mediator between MBASE and the actual theorem proving process. Agents have been designed to query the database for additional knowledge, such as axioms, lemmas, and theorems. More specifically it queries for formulas, that may close actual open subgoals, and reactively suggest these to either the user or the proof-planner.

3.5 Work Program (Goals, Methods, Timeline)

The long-term goal of the project is the development of a mathematical assistance environment and its integration into the emerging Mathematical Semantic Net[2]. Interactive proof systems are today routinely employed in industrial applications for safety and security verification and more recently they find applications in e-learning systems for mathematics. The vision of a powerful mathematical assistance environment which provides computer-based support for most tasks of a mathematician has stimulated new projects and international research networks across the disciplinary and systems boundaries (for an overview see Section 3.3). Furthermore there are now numerous national projects in the US and Europe, which cover partial aspects of this vision, such as knowledge representation, deductive system support, user interfaces, mathematical publishing tools, etc. There exist different frameworks for putting mathematics on the computer, each with a particular aim: proving, checking, calculation, analysis, storage, visualization, etc. However, none of these frameworks is expressive enough to allow the integration of many aims in one system. This also reflects the situation in the unfortunately fragmented deduction systems area and which is similar to that of the AI field as a whole. This was criticised by Nils Nilsson (Kumagai Professor at Stanford, USA) in his speech at IJCAI 2003 where he received the IJCAI Research Excellence Award. Today many of the deduction system subareas even have separate conferences. As a consequence the ambitious goal of an integrated mathematical assistance environments (MASs) was very weakly represented at these conferences and in the deduction systems community until the end of the 90s. It is only very recently that this trend is reversed, with the CALCULEMUS and MKM communities as driving forces of this movement. There exist different strong tools that each address specific tasks arising in mathematical activities. The ΩMEGA system and especially its abstract proof representation used for proof planning and agent-based interactive theorem proving proved to be an ideal environment to integrate and thus combine different tools. In this last phase of the project we aim at further enhancing and extending the ΩMEGA-system to offer support for typical mathematical activities, such as preparation of mathematical publications or tutoring for mathematics students. More specifically, this consists of (i) enhancing the quality of proof development by supporting reasoning directly on the assertion level, (ii) extend the theorem proving environment with a sophisticated maintenance of mathematical knowledge, and (iii) make the structured mathematical knowledge amenable to the theorem proving process by agent-based, reactive suggestion and integration of mathematical knowledge.

[2] `www.win.tue.nl/dw/monet/`

3.5.1 Methods and Work Packages

WP1: Assertion-Level Proof Development

The proofs developed by mathematicians as part of mathematical publications and the proofs developed by students in a mathematical tutoring system are typically developed on an argumentative level. These can be categorized as proofs on the assertion level with different types of underspecifications; a discussion and illustrating examples can be found in (Siekmann et al., 2003) and (Autexier, Benzmüller, Fiedler, Horacek, & Vo, 2003) (the latter examples have been provided by the DIALOG project).
Hence, integrating the mathematical assistant system into the working process of mathematicians requires at least to support the development of proofs directly on the assertion level. The CORE system, which has been recently developed by Autexier (Autexier, 2003) (see Section 3.4) supports proof development directly on the assertion level. The goal of this work package is to completely exchange the natural deduction calculus layer currently underlying the deliberative and reactive proof planning mechanisms in ΩMEGA for the CORE-calculus. This will provide a suitable basis for supporting deliberative and reactive proof planning *directly* on the assertion level. Also underspecification phenomena can be addressed with the help of CORE directly on the proof planning level. This will smoothen the way for employing the mathematical assistant for writing mathematical publications and for tutoring students. In addition it will free the proof planning level completely from the rigidity imposed by the underlying natural deduction calculus, as reported in Section 3.4 (see also (Benzmüller, Meier, Melis, Pollet, & Sorge, 2001) and the report for the project OMEGA (MI4)).
The proposed exchange of the logic layer in ΩMEGA requires the adaptation of all reasoning procedures that are tailored it, including proof planning and the integrations of external systems. WP1 is thus structured as follows: (a) development of an assertion-level interface that is uniform for both the deliberative and reactive planners, (b) adaptation of the integration of external reasoners to return assertion level proofs or proof plans, and (c) development of assertion-level reactive and deliberative proof planning methods and strategies.

WP1.1 Uniform interface for deliberative and reactive planner: In the current stable implementation of ΩMEGA the datastructures for methods used in the deliberative proof planner MULTI and the ΩMEGA-tactics used in the reactive proof planner Ω-ANTS are different. Hence, designed methods could not be used in reactive planning and vice-versa. In the course of exchanging the natural deduction calculus for the CORE-calculus we plan to unify and thus share as much as possible the datastructures between the deliberative and the reactive

planners.
Initial work has already been conducted in that direction in the context of interactive theorem proving on top of CORE. Thereby the Ω-ANTS mechanism has been adapted (Hübner, 2003) to the CORE-calculus in order to reactively suggest and apply possible usable assertions in some proof situation. For the adaptation a so-called task layer (Hübner, Benzmüller, Autexier, & Meier, 2003) was defined on top of CORE, which provides a uniform interface representing the focus of attention together with the usable assertion. Although this task layer was designed for interactive theorem proving on top of CORE, we plan to extend it to a uniform interface for the deliberative and reactive planners. Furthermore, the datastructures underlying deliberative proof-planning methods and the ΩMEGA-tactics underlying the reactive Ω-ANTS-proof planner will be integrated into a single representation of abstract inference steps. Doing so, the proof planning methods of traditional deliberative proof planning on the one hand side and the argument agents of the reactive proof planner will become just two alternative styles of specifying application directions of the *same* abstract inference steps. These will finally enable the combination and even the interleaving of the reactive and deliberative proof-planners.

WP1.2: Integration of external systems: The current system supported the use of external systems contained in the MATHWEB-software bus such as automatic theorem provers, computer algebra system and constraint solvers during the planning process. Calls to these systems could occur at any level, e.g. inside individual methods, inside proof planning strategies, or inside reactive planning agents. Depending on their type these systems returned a proof for some subproblem, a result of some computation, or a value for some constrained variable. Transformation of such results of external systems into ΩMEGA proof objects is possible with tools like SAPPER (Kerber, Kohlhase, & Sorge, 1996) or TRAMP (Meier, 2000) that have been developed in our project. These transformations, of course, strongly depend on the logic layer of the OMEGA system.
In the next phase we plan to redesign the integration of external systems in two aspects: First, we want to move away from the procedural style of calling individual systems via MATHWEB to a declarative style, where only the type of the problem is handled over to the MATHWEB-broker. This mainly results from the redesign of the current MATHWEB-system to integrate limited reasoning capabilities into the brokers of the MATHWEB-system. Second, the external proofs or computations need to be translated in the CORE-calculus instead of the current natural deduction calculus. For the development of such translations we will follow two lines of research: First, similar to the current integration, a proof plan can be created representing the external proof, where individual external proof rules are modelled as abstract inference steps. This proof plan must subsequently be expanded to a calculus level proof. Aside from

a lot of implementation due to the change of the proof planning datastructure this especially requires a technical and qualitative redesign of the tactics expanding the abstract inference step down to the CORE calculus. The second line of research is with respect to resolution and paramodulation based proofs obtained from the external systems. In that case we will investigate whether it is possible to directly translate these type of proofs into CORE derivations. This should be possible in principle, since assertion application in CORE is technically a non-normalform resolution respectively paramodulation inference step. The advantage of having such a translation is that it would overcome the problem of expansion of proof plan steps.

WP1.3: Assertion-level Proof Planning Strategies: The expected benefit of exchanging the natural deduction calculus for the CORE-calculus is that many ΩMEGA-tactics and proof planning methods will be superflous. Especially the tactics/methods that dealt with equation application or unwrapping of hypotheses by goal directed formula decomposition in order to make them applicable will mostly be subsumed by the facilities offered by the underlying CORE calculus. Hence, one part of this work package will consist of purifying the method and tactic collections and end up with these that really are encodings of qualitative mathematical knowledge.

Additionnally, we plan to investigate how the CORE support for directly reasoning on the assertion level can help defining methods and tactics in a more concise manner. More specifically we want to investigate a more abstract representation language inspired by uniform notation for method and tactic premises and conclusions. It should on the one hand allow for a more concise description of the abstract inference patterns and on the other hand make their application less dependent on syntactic equality.

WP2: Maintenance of Mathematical Knowledge

A mathematical proof assistant uses different kinds of knowledge: First, the formalized mathematical domains and problems are organized in structured theories and problems. Second, mathematical proof knowledge encoded in ΩMEGA-tactics, Ω-ANTS-agents, proof planning methods, control knowledge and strategies are exploited; they can be general, theory specific or even problem specific. Maintaining this knowledge explicitly enables the encoding of new mathematical problem solving techniques, such as for instance reasoning about when to integrate which further tactic or axiom. This new line of research is considered in WP3.

When integrating the mathematical proof assistant into typical working activities of mathematicians other types of knowledge will become relevant. For instance if the mathematical assistant is part of a tutor system for students, different sample proofs and proof plans for a same problem need to be maintained in order to use them to advise the students. For supporting the direct preparation of mathematical publications, documents containing both formalised and non-formalised parts need to be related to specific theories, lemmas, theorems, and proofs and changes in these documents need to be supported and maintained. More specifically, for proofs it needs to be checked whether they are still valid after changing parts of a theory, which in turn affects the validity of the mathematical documents. Furthermore, the structuring mechanisms of mathematical theories typically include renaming of mathematical objects, which rises the challenge how specific knowledge such as tactics and methods on the one hand side, but also proofs, proof plans and mathematical documents on the other hand are affected by renamings.

The goal of this workpackage is to develop a tool to maintain the various types of mathematical knowledge. It is based on the experiences with the MAYA-tool (see Section 3.4) that maintains formalized structured theories and especially incorporates a sophisticated management of change to preserve the validity of proofs. The formal structured theories are represented as development graphs, and we plan in a first phase to enrich that representation to include domain specific tactics, methods, control rules, and strategies. The challenging part thereby is to investigate how renamings or general morphisms that occur inside the structured theories can be applied to these objects. Another challenging aspect is to define and represent dependencies between these objects and to extend the truth-maintenance to these objects. For instance, invalidating proofs of theorems that were build using a tactic which is no longer present or which has been changed.

All tasks proposed in this workpackage are oriented towards adapting software engineering techniques for knowledge management to the specific area of mathematical knoweldge. Hence we expect that any progress in that workpackage directly contributes to the mathematical knowledge management research challenge, which is an emerging world-wide topic not only in mathematics (e.g. mathematical journals) but especially in domains relying on mathematical descriptions as for example physics.

WP3: Integration of Mathematical Knowledge

When starting the proof of some mathematical problem in a particular mathematical domain, for resource reasons typically not *all* relevant knowledge represented in the mathematical assistant and the mathematical knowledge repository is handed over to the proof planner but only parts of it. Thus, an initial selection of the available knowledge is passed to the planner, which currently consists for each problem of a set of axioms, tactics and proof-planning methods. As this selection may be incomplete, there is a need to incrementally incorporate additional knowledge if it is required, i.e. if the available knowledge ressources are insufficient for a successful generation of a proof plan.

Mathematical knowledge occurs in the mathematical assistant at different places and there is a need to access that knowledge. Respective knowledge occurs, for instance, in the partial proof object of some actual problem (e.g. hypotheses), or within ΩMEGA-tactics and proof planning methods. This knowledge must be accessible and, furthermore, it must checked whether it is applicable and what the resulting proof situation, i.e. proof object, would be. On a qualitative level, it must be analysed whether the application of this knowledge is useful for the proof progress. These three types of knowledge can be queried already, namely the CoRE-system that provides directly information about applicable hypotheses, the Ω-ANTS-system for ressource-adaptive computation of applicable ΩMEGA-tactics and the proof planner MULTI for deliberative selection and application of proof-planning methods. This includes the use of external systems like computer algebra or automatic theorem provers, which can be onvoked from inside the agents, inside ΩMEGA-tactics or proof-planning methods.

The other place where knowledge occurs is in the structured mathematical knowledge base (see workpackage WP2). This perspective enables the development of new kinds of mathematical problem solving techniques: First, exploiting the explicit structure of the proof knowledge not only allows to statically determine the visible axioms and lemmas for a given problem, but also the domain specific tactics, agents, methods and control rules can be provided to the reactive or deliberative planners. This opens a yet not considered line of research, namely reasoning about which type of knowledge should be integrated when into the reasoning process. Second, the structure of the theories can be exploited to investigate lemma speculation techniques, where the structure of the theories is exploited to speculate lemmas inside the right theories. All these aspects will allow to move from a so far passive database with static knowledge towards an active reasoning system that exploits the structured mathematical knowledge to assist the reactive and deliberative planners. To this end we plan to investigate how the reactive Ω-ANTS-system can be exploited to design special purpose reactive agents that work on the structure of the mathematical knowledge base that pro-actively generate useful mathematical knowledge for the actual planners.

The research proposed in this workpackage is strongly connected to the research objectives of the Calculemus community, which seeks the development of methods for a coordinated integration and cooperation of different knowledge sources, namely computer algebra systems, theorem proving systems, but also mathematical knowledge bases.

WP4: Evaluation

This workpackage is concerned with the evaluation of the techniques developed in the previous workpackages, especially with respect to the objective to integrate the mathematical assistant into typical mathematical activities. More specifically we will eveluate the developed solutions by (1) using the proof assistant to develop proofs in a form as they are written in mathematical publications and (2) use the proof assistant as part of a mathematics tutoring system.

WP4.1: Development of Textbook Style Proofs. For the evaluation we plan to develop proofs in the mathematical assistant as close as possible to the format as they are presented in a standard mathematical textbook (Dieudonné, 1969). This requires on the one hand side the formalisation of the mathematical objects and theories in the mathematical knowledge base. For the proofs of the lemmas and theorems appropriate abstract inference steps must be modelled that enable the representation of the textbook proofs as heterogenous proof plans build from abstract inference steps and assertion applications. The automatic validation of that proof plan by expansion to the CoRe-calculus level then requires the definition of the respective strategies, methods and agents.

WP4.2: Mathematics Tutoring. This evaluation will take place in close cooperation with the SFB project DIALOG. Thereby we plan to use the proof assistant as a part of the tutoring system, that deals with the formal analysis of the partial proofs obtained after linguistic analysis and disambiguation. The resulting proofs are represented again as (partial) proof plans. These can either be compared to given proof plans and provide the appropriate feedback to the proof manager (see description of Project MI3 DIALOG). Alternatively, for instance if no matching proof could be found, the proof planning systems can be used to try to complete the partial proof plan. If this succeeds, the proof manager can be informed that the student is doing a proof that was not known beforehand. If the proof planner fails to complete the proof plan, it can inform the proof manager accordingly.

The evaluations described in this section are also directly relevant to the mathematical knowledge management community (MKM). It is planned to compare the proofs performed in the proof assistant with the same proofs performed in the THEOREMA-tool of Prof. B. Buchberger, with whom we agreed to use the mathematical textbook of Dieudonné (Dieudonné, 1969).

3.5.2 Timeline

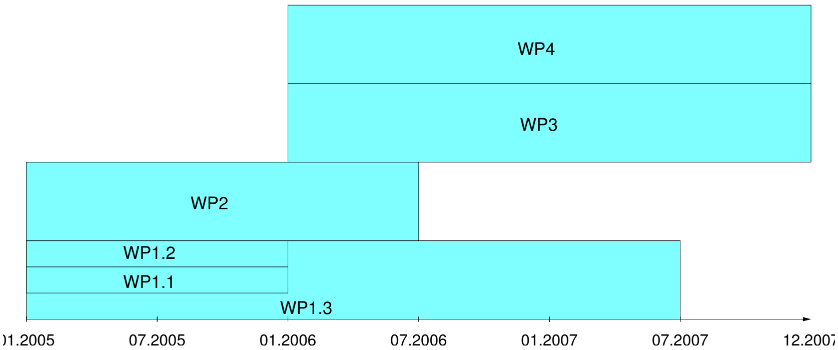

Figure 1. Timeline of the workpackages

3.6 Position Within the Collaborative Research Center

3.7 Differences From Other Funded Projects

References

Abel, A., Chang, B.-Y. E., & Pfenning, F. (2001). Human-readable machine-verifiable proofs for teaching constructive logic. In U. Egly, A. Fiedler, H. Horacek, & S. Schmitt (Eds.), *Proceedings of the Workshop on Proof Transformations, Proof Presentations and Complexity of Proofs (PTP'01).* Universitá degli studi di Siena.

Andreoli, J.-M. (2000). Focussing and proof construction. *Annals of Pure and Applied Logic*, *107*(1), 131-163.

Audebaud, P., & Rideau, L. (2003). Texmacs as authoring tool for publication and dissemination of formal developments. In *Workshop on User Interface for Theorem Provers (UITP03).* Rome.

Autexier, S. (2003). *Hierarchical contextual reasoning.* Unpublished doctoral dissertation, Computer Science Department, Saarland University, Saarbrücken, Germany.

Autexier, S., Benzmüller, C., Fiedler, A., Horacek, H., & Vo, B. Q. (2003). Assertion-level proof representation with under-specification. *Electronic in Theoretical Computer Science*. (To appear)

Autexier, S., Hutter, D., Mossakowski, T., & Schairer, A. (2002). The development graph manager MAYA. In H. Kirchner & C. Ringeissen (Eds.), *Proceedings 9th International Conference on Algebraic Methodology And Software Technology (AMAST'02)* (Vol. 2422). Springer.

Autexier, S., & Mossakowski, T. (2002). Integrating HOL-CASL into the development graph manager MAYA. In A. Armando (Ed.), *Proceedings of FROCOS'02* (Vol. 2309, pp. 2–17). Springer.

Benzmüller, C., Cheikhrouhou, L., Fehrer, D., Fiedler, A., Huang, X., Kerber, M., Kohlhase, M., Konrad, K., Melis, E., Meier, A., Schaarschmidt, W., Siekmann, J., & Sorge, V. (1997). ΩMega: Towards a Mathematical Assistant. In W. McCune (Ed.), *Proceedings of the 14th International Conference on Automated Deduction (CADE–14)* (Vol. 1249, pp. 252–255). Townsville, Australia: Springer Verlag, Germany.

Benzmüller, C., Giromini, C., Nonnengart, A., & Zimmer, J. (2002). Reasoning services in the mathweb-sb for symbolic verification of hybrid systems. In *Proceedings of the Verification Workshop - VERIFY'02 in connection with FLOC 2002.* København.

Benzmüller, C., Meier, A., Melis, E., Pollet, M., & Sorge, V. (2001). Proof planning: A fresh start? In *Proceedings of the IJCAR 2001 Workshop: Future Directions in Automated Reasoning* (pp. 25–37). Siena, Italy.

Bertot, Y., Guilhot, F., & Pottier, L. (2003). Visualizing geometrical statements with geoview. In *Workshop on User Interface for Theorem Provers (UITP03).* Rome.

Buchberger, B., Jebelean, T., Kriftner, F., Marin, M., Tomuta, E., & Vasaru, D. (1997). A survey of the theorema project. In *International Symposium on Symbolic and Algebraic Computation* (p. 384-391).

Bundy, A., Harmelen, F. van, Hesketh, J., & Smaill, A. (1991). Experiments with proof plans for induction. *Journal of Automated Reasoning*, *7*, 303–324.

Bundy, A., Harmelen, F. van, Horn, C., & Smaill, A. (1990). The Oyster-Clam System. In M. Stickel (Ed.), *Proceedings of the 10th International Conference on Automated Deduction (CADE–10)* (Vol. 449, pp. 647–648). Kaiserslautern, Germany.

Bundy, A., Harmelen, F. van, Ireland, A., & Smaill, A. (1990). Extensions to the rippling-out tactic for guiding inductive proofs. In M. Stickel (Ed.), *Proceedings of the 10th International Conference on Automated Deduction (CADE–10)* (Vol. 449, pp. 132–146). Kaiserslautern, Germany.

Caprotti, O., & Schreiner, W. (November, 2002). *Mathbroker overview* (Tech. Rep.). Project Report, RISC-Linz, Johannes Kepler University, Linz, Austria.

Cohen, A., Murray, S., Pollet, M., & Sorge, V. (2003). Certifying solutions to permutation group problems. In F. Baader (Ed.), *19th Conference on Automated Deduction (CADE-19)* (Vol. 2741, pp. 258–273). Springer.

DeBruijn, N. G. (1970). The mathematical language AUTOMATH, its usage and some of its extensions. In *Symposium on Automatic Demonstration* (p. 29-61).

Dennis, L. A., Collins, G., Norrish, M., Boulton, R., Slind, K., Robinson, G., Gordon, M., & Melham, T. (2000). The prosper toolkit. In *Proceedings of the 6th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (Vol. 1785). Berlin, Germany: Springer Verlag.

Dieudonné, J. A. (1969). *Foundations of modern analysis* (Vol. 10-I). New York and London: Academic Press.

Dixon, L., & Fleuriot, J. (2003). ISAPLANNER: A prototype proof planner in isabelle. In 19^{th} *International Conference on Automated Deduction (CADE).* Springer.

Fiedler, A. (2001a). *User-adaptive proof explanation.* Unpublished doctoral dissertation, Naturwissenschaftlich-Technische Fakultät I, Saarland University, Saarbrücken, Germany.

Fiedler, A. (2001b). P.rex: An interactive proof explainer. In R. Goré, A. Leitsch, & T. Nipkow (Eds.), *Automated Reasoning — 1st International Joint Conference, IJCAR 2001* (Vol. 2083, pp. 416–420). Siena, Italy: Springer.

Franke, A. (2003). *Inhaltsorientierte verwaltung mathematischen wissens.* diploma thesis, Computer Science Department, Saarland University, P.O. Box 15 11 50, 66041 Saarbrücken.

Franke, A., & Kohlhase, M. (1999). System description: MATHWEB, an agent-based communication layer for distributed automated theorem proving. In H. Ganzinger (Ed.), *Proceedings of the 16th Conference on Automated Deduction* (pp. 217–221). Springer Verlag.

Franke, A., & Kohlhase, M. (2000). System description: MBASE, an open mathematical knowledge base. In D. McAllester (Ed.), *Automated Deduction – CADE-17* (pp. 455–459). Springer Verlag.

Gordon, M. J. C., & Melham, T. F. (1993). *Introduction to HOL – a theorem proving environment for higher order logic.* Cambridge University Press.

Huang, X. (1994a). *Human oriented proof presentation: A reconstructive approach.* Unpublished doctoral dissertation, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany.

Huang, X. (1994b). Reconstructing proofs at the assertion level. In A. Bundy (Ed.), *Proceedings of the 12th Conference on Automated Deduction* (pp. 738–752). Nancy, France: Springer Verlag.

Huang, X., Kerber, M., Kohlhase, M., Melis, E., Nesmith, D., Richts, J., & Siekmann, J. (1994). Ω-MKRP: A Proof Development Environment. In A. Bundy (Ed.), *Proceedings of the 12th International Conference on Automated Deduction (CADE–12)* (Vol. 814, pp. 788–792). Nancy, France: Springer Verlag, Germany.

Hübner, M. (2003). *Interactive Theorem Proving with Indexed Formulas.* Unpublished master's thesis, Department of Computer Science, Saarland University.

Hübner, M., Benzmüller, C., Autexier, S., & Meier, A. (2003). Interactive proof construction at the task level. In *In Proceedings of the Workshop User Interfaces for Theorem Provers (UITP 2003).* Rome, Italy.

Hutter, D. (1990). Guiding induction proofs. In M. Stickel (Ed.), *Proceedings of the 10th Conference on Automated Deduction* (p. 147-161). Kaiserslautern, Germany: Springer Verlag.

Hutter, D. (2000). Management of change in structured verification. In *Proceedings of Automated Software Engineering, ASE-2000.* IEEE.

Kamareddine, F., & Nederpelt, R. (2004). A refinement of de bruijn's formal language of mathematics. *Logic, Language and Information.*

Kerber, M., Kohlhase, M., & Sorge, V. (1996). Integrating Computer Algebra with Proof Planning. In J. Calmet & C. Limongelli (Eds.), *Design and Implementation of Symbolic Computation Systems; International Symposium, DISCO '96, Karlsruhe, Germany, September 18-20, 1996; Proceedings* (Vol. LNCS1128, pp. 204–215). Berlin;Heidelberg;New York: Springer.

Kohlhase, M. (2000). OMDOC: An infrastructure for OPENMATH content dictionary information. *Bulletin of the ACM Special Interest Group on Symbolic and Automated Mathematics (SIGSAM)*, *34*(2).

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 (CADE–17)* (Vol. 1831, p. 460-464). Pittsburgh, USA: Springer Verlag, Berlin, Germany.

Meier, A. (2004). *Proof planning with multiple strategies.* Unpublished doctoral dissertation, Computer Science Department, Saarland University, Saarbrücken, Germany.

Meier, A., Pollet, M., & Sorge, V. (2001). Exploring the Domain of Residue Classes. In B. Buchberger (Ed.), *Workshop on Computer Algebra and Automated Theorem Proving: Interactions, held on the Eight International Conference on Computer Aided Systems Theory (EUROCAST 2001).*

Melis, E., & Siekmann, J. (1999). Knowledge-based proof planning. *Artificial Intelligence*, *115*(1), 65-105.

Melis, E., & Ullrich, C. (2001). How to teach it – polya-inspired scenarios in activemath. *AI in Education (AIED-2003), IOS Press*, 141-147. (www.activemath.org)

Mossakowski, T., Autexier, S., & Hutter, D. (2001). Extending development graphs with hiding. In H. Hussmann (Ed.), *Proceedings of Fundamental Approaches to Software Engineering (FASE 2001)* (p. 269-283). Genova: Springer.

Richardson, J., Smaill, A., & Green, I. (1998). System description: Proof planning in higher-order logic with λ*Clam*. In C. Kirchner & H. Kirchner (Eds.), *Proceedings of the 15th International Conference on Automated Deduction (CADE–15)* (Vol. 1421, pp. 129–133). Lindau, Germany: Springer Verlag, Germany.

Scheines, R., & Sieg, W. (1993). In J. V. Boettcher (Ed.), *101 success stories of information technology in higher education: The joe wyatt challenge.* McGraw-Hill.

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 five years of automating mathematics* (p. 271-314). Kluwer Academic Publishers. (ISBN 1-4020-1656-5)

Siekmann, J., Benzmüller, C., Fiedler, A., Meier, A., & Pollet, M. (2002). Proof development with Omega: Sqrt(2) is irrational. In M. Baaz & A. Voronkov (Eds.), *Logic for Programming, Artificial Intelligence, and Reasoning, 9th International Conference, LPAR 2002* (Vol. 2514, pp. 367–387). Springer.

Team, T. C. D. (2003). *The coq proof assistant reference manual (version 7.4)* (Tech. Rep.). INRIA, Rocquencourt, France.

Zimmer, J. (2004). A Framework for Agent-based Brokering of Reasoning Services. In R. Monroy, G. A. Figueroa, & L. E. Sucar (Eds.), *Proceedings of the Mexican International Conference on Artificial Intelligence 2004.* Springer-Verlag. (to appear)

Zimmer, J., Armando, A., & Giromini, C. (2001). Towards mathematical agents — combining mathweb-sb and LBA. In *Proceedings of the 9th CALCULEMUS Symposium 2001* (pp. 21–23). Siena (Italy).

Zimmer, J., & Dennis, L. (2002). Inductive theorem proving and computer algebra in the mathweb software bus. In *Proceedings of the 10th CALCULEMUS Symposium 2002.* Marseille (France).

Zimmer, J., Franke, A., Colton, S., & Sutcliffe, G. (2002). Integrating hr and tptp2x into mathweb to compare automated theorem provers. In *Proceedings of the PaPS'02 Workshop* (pp. ?–?).

Zimmer, J., & Kohlhase, M. (2002). System description: The MathWeb software bus for distributed mathematical reasoning. In A. Voronkov (Ed.), *18th Conference on Automated Deduction (CADE-18)* (Vol. 2392). Copenhagen, Denmark: Springer.

3.8 External Funding for the Project MI 4

PC: Personnel requirements and costs (see 3.8.1 for explanation)
AC: Administrative costs (see 3.8.2 for explanation)
I: Investments (equipment exceeding 10K EUR; see 3.8.3 for explanation)

	Grant 2004			2005			2006			2007		
PC	Salary Group	No.	Amount EUR	Salary Group	No.	Amount EUR	Salary Group	No.	Amount EUR	Salary Group	No.	Amount EUR
	Bat IIa	2	112.800	Bat IIa	2	112.800	Bat IIa	2	112.800	Bat IIa	2	112.800
	SHK	3	54.000	SHK	3	54.000	SHK	3	54.000	SHK	3	54.000
	Total:	5	166.800	Total:	5	166.800	Total:	5	166.800	Total:	5	166.800
AC				Expense category or number		Amount EUR	Expense category or number		Amount EUR	Expense category or number		Amount EUR
				547		5.000	547		5.000	547		5.000
				Total:		5.000	Total:		5.000	Total:		5.000
I				Total investments			Total investments			Total investments		
				0			0			0		

3.8.1 Explanation of Personnel Requirements

	Name, degree, position	Specific field of the employee	Institute of the university or other organization	Contribution to the project in hours/week (consulting: C)	In this position in the CRC since	Proposed BAT level
University funding						
3.8.1.1 Scientific personnel (including assistants)	1. Jörg Siekmann, Prof. Dr., full professor 2. Serge Autexier, Dr., researcher 3. Christoph Benzmüller, Dr., assistant professor	Computer Science Computer science Computer science	Uni., FR Informatik DFKI GmbH Uni., FR Informatik	5 10 10	1.1.1996 1.1.2001	– – –
3.8.1.2 Nonscientific personnel	4. Irmtraud Stein, Verwaltungsangestellte	–	Uni., FR Informatik	5	1.1.1996	–
External funding						
3.8.1.3 Scientific personnel (including assistants)	5. Martin Pollet 6. Quoc Bao Vo 9. Marc Wagner 10. NN	Mathematics Computer Science Computer Science Computer Science	Uni., FR Informatik Uni., FR Informatik Uni., FR Informatik Uni., FR Informatik	38,5 38,5 19 19	??? ??? ???	BAT IIa BAT IIa SHK SHK
3.8.1.4 Nonscientific personnel						

(Positions for which funding is being applied for *for the first time* are marked with X.)

Mitarbeiter durchnummerieren und Aufgabenbeschreibung nachfolgend erläutern.

Bitte Verfahrensgrundsätze der Deutschen Forschungsgemeinschaft zur Bezahlung wissenschaftlicher Mitarbeiter beachten.

Job Descriptions of University Personnel

1. Jörg Siekmann: project leader
2. Serge Autexier: project leader
3. Christoph Benzmüller: project leader
4. Irmtraud Stein: Secretary

Job Descriptions of Externally Funded Personnel

5. Martin Pollet: (WP1) (WP3) (WP4)
6. Quoc Bao Vo: (WP2) (WP3) (WP4)
7. Jessi Berkelhammer : WP4
8. Syed Hussein : WP1, WP3
9. Marc Wagner : WP4, WP2
10. NN :

3.8.2 Specification and Explanation of Administrative Costs (by Fiscal Year)

	2005	2006	2007
For administrative expenses, the following amounts are expected to be available from the *university's own funds*:	1000	1000	1000
For administrative expenses, the following amounts are requested as *external funding* (corresponding with the total amounts under "administrative expenses" in the overview 3.8):	4000	4000	4000

(All amounts in EUR.)

Explanation of the Requested *External Funding* for Administrative Expenses

(547) Other Expenses

The annual university funds for the chair of Prof. Siekmann are 2856€. The DFG currently funds the SFB-projects OMEGA and DIALOG as well as the individual project FABEON. For each of these projects the DFG assumes 500 – 1000€
...macht 54 Euro fehlbetrag!!

3.8.3 Investments (Equipment With Gross Cost Above 10.000 EUR and Vehicles)

	Requested for the fiscal year		
	2005	2006	2007
Amount:			

(All prices in EUR *including* VAT, transportation costs, etc.)

Explanation of the Requested External Funding for Investments

Bitte im folgenden – nach Haushaltsjahren getrennt – die beantragten Geräte einzeln begründen. Geben Sie hierbei für jedes Gerät die Bezeichnung des Gerätes, (ggf. Typenbezeichnung), die Antragssume in Euro und die Begründung.