An Interactive Proof Development Environment + Anticipation = A Mathematical Assistant?

Jörg Siekmann, Helmut Horacek, Michael Kohlhase,
C. Benzmüller, L. Cheikhrouhou, D. Fehrer, A. Fiedler,
S. Hess, K. Konrad, A. Meier, E. Melis, V. Sorge

FB Informatik, Universität des Saarlandes
D-66041 Saarbrücken, Germany
fax:(49) 681-302-2235
omega@ags.uni-sb.de, http://www.ags.uni-sb.de/

Abstract
Current semi-automated theorem provers are often advertised as “mathematical assistant systems”. However, these tools behave too passively and in a stereotypic way to meet this ambitious goal because they lack the capability to adequately take into account requirements on proof search control and user demands for their own actions. Motivated by this deficit, we have incorporated several facilities into the ΩMEGA proof development system that anticipate a number of divergent factors, based on mathematical knowledge, proof search defaults, and expectations about users. The techniques enhance the system’s functionality through proof planning by knowledge-intensive methods, proof search guidance by default suggesting agents, and proof presentation by redundancy avoidance measures. The system’s behavior suggests that anticipation is without doubt a central driving force in a mathematical assistant.

Keywords: Mathematical Assistant System, Automated Theorem Proving, Proof planning, Agents, Proof Presentation.

1 Introduction

Mechanized reasoning systems are built with various purposes in mind. One goal is the construction of an autonomous theorem prover (ATP), whose strength achieves or even surpasses the ability of human mathematicians. Another is to build a system where the user derives the proof, with the system guaranteeing its correctness (an interactive theorem prover: ITP). A third purpose consists in modeling human problem-solving behavior on a machine, that is, cognitive aspects are the focus (human-oriented (plan-based) theorem prover: HTP).

Advanced theorem proving systems often try to combine the different goals, since they can complement each other in an ideal way: ATP can routinely solve simple problems, but for difficult problems suffer from limited domain knowledge, and their search space is typically too large since they cannot take high-level control knowledge into account. ITP and HTP aim at overcoming these problems from different perspectives. ITP by giving the user the possibility to guide the proof search and to directly import expert knowledge into the system, while plan-based mechanisms (HTP) directly formalize method- and control knowledge and thus make it available to (automated) proof search. Despite a number of achievements (Cheikhrouhou & Siekmann, 1998; Melis, 1998; Bundy, van Harmelen, Hesketh, & Smaill, 1991), the functionality of HTP still proves to be insufficient, in comparison with the support a human expert could give.

In our own ΩMEGA system (Benzmüller et al., 1997) – a plan-based HTP – it is explicitly tried to combine the reasoning power of ATP as logic engines, the specialized problem solving knowledge of the proof planning mechanism (from HTP), and the interactive support of ITP. We think that the combination of these complementary approaches inherits more advantages than drawbacks, because for most tasks domain-specific as well as domain-independent problem-solving know-how is required and for difficult tasks more often than not an explicit user-interaction should be provided. While this combination of approaches (let us call such systems *Interactive Proof Development Systems* IPDE) gives more computer support to a user than single-purpose systems, it does not lead to a system that behaves like a competent mathematical assistant. As a tendency, IPDE act too passively and in a stereotypic way, because their control knowledge is still fairly limited, and significantly augmenting it through user interaction is time-consuming and may be laborious, e.g. due to inadequate presentation facilities.

Motivated by these deficits, we have incorporated facilities into the ΩMEGA IPDE that anticipate a number of divergent factors, based on mathematical knowledge, proof search defaults, and expectations about users. The techniques enhance the system's functionality through proof planning by knowledge-intensive methods, proof search guidance by default suggesting agents, and proof presentation by redundancy avoidance measures and make the system behave more like an assistant system. Consequently, we argue in favor of anticipation as a major driving force in promoting an IPDE into a mathematical assistant.

In the following, we will discuss this claim emphasizing relations to the role of anticipation in rational behavior in general. First we will illustrate shortcomings of IPDE in more detail. Then we discuss the role of anticipation in the interactive proof development environment ΩMEGA, in relation to the general perspective of guiding rationality-driven processes in an effective way. In the central parts of the paper, we elaborate the manifestations of anticipation in ΩMEGA, which plays a central role in three facets: control knowledge, interaction knowledge, and presentation knowledge.

2 Shortcomings of Interactive Proof Development Environments

The design of interactive proof development environments is motivated by various insights in the limitations of fully automated theorem provers. The basic idea is that a mathematician in his developing of a large proof can be supported by a computer system to which he can delegate some sorts of subproblems, especially laborious and moderately complicated ones. In practice, however, it has frequently turned out that this simple looking distribution of labor is difficult to obtain, so that interactive proof development environments appear insufficient for rendering the work of a mathematician effective. In our view, this situation is caused by some serious deficits:

- The level of abstraction of the logical calculus of the current theorem provers is inadequate for influencing proof strategies in a knowledgeable way, in particular, the specification facilities differ significantly from the vocabulary mathematicians are familiar with.
- Control mechanisms for guiding the proof search are widely restricted to static parameters that capture dynamic aspects of the proof state in an insufficient manner and do not allow interactive additions and modifications.
- Facilities for specifying the problem to be proved and for presenting the results obtained are generally inconvenient for users. In particular, directly converting the output of a theorem prover into natural language leads to presentations that users consider to be redundant in large parts.

It is our firm belief that improving this situation requires a system to make better use of communicative resources by exploiting defaults and expectations of various sorts. Doing this means taking potential future actions and states into account in determining its actions, which involves *anticipation* in a broad sense. In the following, we clarify our position towards this concept from a general perspective as well as from the specific view of interactive proof development environments.

3 Manifestations of Anticipation in ΩMEGA

Anticipation is an important concept underlying many aspects of humans' rational behavior, especially when the task involves choosing among several potentially useful actions. Rather than merely relying on properties of the current situation and associated preferences, the additional thoughts and effort invested are likely to improve the decision quality in view of at least medium term considerations. In this sense, anticipation constitutes some sort of foresight which is characterized by consultation of knowledge and awareness of the associated uncertainties.

The actions reasoned about may be physical actions for achieving a long-term goal, deciding about the sort of the action to be performed (a physical or interactive one), and tailoring communicative actions. We will now relate these general categories of anticipation and their particularities with their manifestation in the ΩMEGA system introduced above.

1. Reasoning about the relative merits of *physical actions* is characterized by comparing additional selection efforts with later gain through more economic goal satisfaction. For example, buying a simple object such as a pen is usually done very quickly, while buying larger and more expensive objects such as a car makes detailed comparisons for price as well as for technical details well worth the additional effort.

   In ΩMEGA, the proof planner constructs a proof plan for the goal node from a set of supporting nodes (the proof assumptions) using a set of proof planning operators, called methods, whose contextual usefulness is judged by so-called control rules (Melis, 1990). The control rules associated with a method *anticipate the suitability of this method* for the problem at hand. Here, method application corresponds of a physical action, since it changes the current proof state.

2. Reasoning about the relative merits of *inter-actions* is characterized by comparing the additionally necessary effort for communication to the expected gain through the help of the agent addressed. For example, a request for a route description is usually easy to communicate, while a request for writing a letter may require to add too much background information, so that letters are typically written by the initiators themselves.

   ΩMEGA uses agents with specific proof technique expertise to enhance control over proof search. These agents are part of a multi-layered focusing mechanism for computing suitable default values supporting the application of inference rules in a proof state (AIMSA98, 1998). The agents *anticipate the usefulness of certain inter-actions* in dependency of properties of the current proof state.

3. Reasoning about the relative merits of *communicative actions* is characterized by comparing additional communicative efforts with the expected gain of comfort for subsequent actions. For example, providing surplus information such as the track to the departure time typically saves the time needed for follow-up questions, while adding redundant information may lead to confusion rather than to emphasis.

   ΩMEGA uses an extension of the PROVERB system (Huang & Fiedler, 1997) that presents proofs in natural language. In order to emphasize concise texts that resemble those found in mathematical textbooks, PROVERB employs a small number of contextually motivated rules (Horacek, 1998) expressing aspects of conversational implicature. These rules *anticipate the addressee's inferences* in adapting the proof output to the user's needs.

In the following sections, we discuss each of the facets of anticipation incorporated in the system so far, according to the above item list.

3.1 Anticipation in Control

Methods in ΩMEGA can be seen as plan operators that manipulate the planning state of a proof. The plan structures are expandable into a formal proof in Natural Deduction calculus. Methods may range from very general proof techniques such as diagonalization (Cheikhrouhou & Siekmann, 1998), over more restricted ones, such as methods for proving the Heine-Borel theorems (Melis, 1998), to the incorporation of external problem solvers, such as computer algebra systems and constraint solvers (Kerber, Kohlhase, & Sorge, 1998). The adequate application of methods from each of these areas requires control knowledge. Some pieces of this type of knowledge are rather general, while others refer to specific aspects of the respective area. There are two complementary organization principles underlying the representation of control knowledge:

1. control knowledge represented in a specific method, and
2. control knowledge that is separately represented, which encapsulates search heuristics.

The control knowledge represented in methods comprises the legal and local conditions for the application of the method. Legal, as opposed to heuristic, means that without the condition the method fails to be applicable, e.g., an appropriate instantiation of the method parameters or a certain relation between the formulas that are manipulated and used by the method. Local, as opposed to global, means that the knowledge can only describe properties and relations of the formulas processed by the method.

The control knowledge in control-rules represents heuristics for the decisions that occur in planning, that is, decisions about potentially suitable methods, about the order in which goals should be satisfied, and about instantiations of method parameters. ΩMEGA's control-rules allow for preferring a decision over alternatives, for rejecting a choice, and for determining a particular choice. The last two kinds of control-rules prune the search space. Control-rules can capture global knowledge, as e.g., relations between methods, knowledge on how a method is used to prove a theorem in a particular theory, knowledge on how the planning history including failed attempts changes the planning preferences, how resources influence the choice of that method, etc.

An example for decisions of the choice of a method, is given in the theory for planning limit theorems. Here, simple inequalities may be satisfied by a method that performs a direct estimation and more complex inequalities should rather be proved by a method that performs a decomposition to simpler inequalities.

As for the goal decisions, the reordering of the relative priority in which subgoals are to be satisfied, can be based on patterns in which meta-variables appear, that is, symbols that will eventually be instantiated to predicates, functions, or formulae. If, for example, such a meta-variable appears as a head in a goal g_1 that is first in a goal agenda and the same meta-variable appears in a goal g_2 that comes later in the agenda, then the order priorities are changed. The idea is that goals associated with more restrictions should be satisfied first, and a meta-variable appearing at an interior position of a formula is always associated with more restrictions than a meta-variable appearing as the head of some formula because the type restrictions of the predicate that dominates the meta-variable at an interior position of a formula additionally restrict potential instantiations of that meta-variable.

3.2 Anticipation in Interaction

The acceptance and usability of interactive theorem proving environments is, among other things, strongly influenced by the availability of an intelligent default suggestion mechanism for proof search commands. Such mechanisms support the user by decreasing and simplifying the necessary interactions during the proof construction process. Since the user of a tactic-based theorem prover has the choice between many commands in each proof step, it is desirable to preselect those commands that are meaningfully applicable in view of the given proof state and history. Moreover, heuristics can be employed in order to promote commands that are most likely to contribute to a proof of a subproblem in focus.

In ΩMEGA, we realize these ideas by a new blackboard mechanism for guiding interactive proofs (AIMSA98, 1998). Within this, we combine an agent architecture to compute suggestions with a two layered focus technique that structures a partial proof.

1. Autonomous agents are employed to gather information on possible argument suggestions for commands from a partial proof. In a first layer of agents, each agent is related to a single command parameter, and it contains specific structural and logical information as requirements for valid arguments of this parameter. Depending on the type of the parameter it either uses this information to search for corresponding nodes in the proof tree or to directly compute its suggestion. Agents communicate already retrieved arguments to one another via a blackboard that cummulates all suggestions. This enables other agents to incorporate information found on the blackboard in their own search process, thereby increasing the quality of their suggestions. A second layer of agents monitors the argument suggestion blackboards for several commands and combines this information in order to compute command suggestions. Commands are suggested only when at least one argument can be instantiated. Furthermore, commands are sorted with respect to heuristic criteria such as: Can a subproblem be completely solved by a command application? Which command can contribute to the simplification of an open problem? Which command is most likely to be applied by the user in view of the proof state and history?

2. Heuristic information and guidance for the agents is provided by focus structures that partition a proof tree according to two organization principles: chronological order and logical dependencies. One of the structures sorts proof lines chronologically in the order in which they have been created. Thus agents always consider newly derived lines first in their search process. The other structure makes explicit

logical dependencies of proof lines that are only implicitly given in the proof tree, which is done by grouping open subproblems together with relevant support lines. This enables argument agents to restrict their search to partial proofs relevant for a focused subproblem.

In combination, both mechanisms enable the system to suggest commands in a way that anticipates proof steps that are likely to be chosen by a human user. For this anticipation process, we assume a human problem solving behavior where proof steps are derived when they are needed to contribute to the solution, and subgoals are proved in a consecutive fashion, that is, one open subproblem is closed before considering the next. Yet the user has still the possibility to freely choose commands other than any suggested ones. Moreover the suggestion mechanism can be adapted in a way that a user can easily introduce new agents for any given command or modify existing ones. Beyond that, the behavior of the suggestion mechanism can be tailored to a user's preferences by assigning resources and ratings to agents, thereby influencing the computational behavior and efficiency of the mechanism.

3.3 Anticipation in Presentation

Once a subproof is found in the ΩMEGA system (by an ATP or the proof planner), this result needs to be communicated in a user-adequate form. Hereby, the role of presentation knowledge is to control the manipulation of the output of a theorem prover accordingly. The resulting presentation should not only convey the entire content of the proof but it should also meet the expertise and reasoning skills of the addressee. Consequently, presentations are envisioned that resemble most closely the style of proofs as typically found in mathematical text books.

The main aspect that distinguishes theorem prover output from text book presentations is the amount of information and the resulting degree of detail in which proofs are documented: a sequence of partially trivial details in machine-found proofs, as opposed to concise presentations in text books. Consequently, many of the details contained in documentations of machine-found proofs must be left out in the ultimately presented texts, because they appear redundant and even distracting to humans. Confronted with concise presentations of the remaining parts, humans are then assumed to be able to mentally reconstruct the underlying picture to the required degree of detail, so that some of the impacts of human capabilities in understanding proofs are anticipated.

In order to enable a system to produce such concise presentations, every effort must be undertaken to identify those details which – in the given context – appear redundant to the reader. Such a system requires control over a model that incorporates specific forms of human reasoning and a mechanism for producing proof presentations in view of this user-oriented model. In the following, we expose the major ingredients of those parts of ΩMEGA's proof presentation model which are responsible for conciseness of presentations. The model of human reasoning comprises the following categories of mental capabilities:

1. taxonomic and referential knowledge,
2. taxonomic and logical inference capabilities,
3. the attentional state, with respect to the current discourse segment, and
4. communicatively motivated inferences.

The first three categories aim at assessing whether or not the human addressee can reasonably be assumed to be able to infer some proposition needed for understanding a presentation from a set of already known propositions. Herein, the taxonomic knowledge essentially contributes acquaintance with axioms, according to underlying mathematical theories. The inference capabilities are tailored to domain-specific reasoning skills, such as mentally performing logical substitutions and chaining of elementary inference steps, such as generalizations and instantiations, up to a certain complexity, oriented on assertions mentioned or omitted in corresponding text book presentations. Finally, the attentional state imposes restrictions on the accessibility of discourse entities, such as the scope of some subproof partitions, thereby reflecting human memory limitations.

The capability under the fourth category comprises consequences from the other three on the overall presentation. The associated knowledge is encapsulated in a small set of rules which contain references to the required pieces of knowledge and inferential skills. These rules embody the kernel of the mechanism that is responsible for introducing presentation-motivated modifications in the documentation of proofs, namely:

1. Omitting trivial propositions, such as $0 < 1$.
2. Omitting domain regularities that justify an inference step, such as associativity and commutativity justify the derivation of an expression with permuted operations from an otherwise identical one.
3. Omitting an intermediate inference step, if the intermediate and the ultimately justifying propositions are structurally similar, such as $0 < a$ and $1 < a$. This is a domain-specific criterion to model presentation coherence.

Application of these rules leads to the introduction of short-cuts and reorganizations in the original proof documentation – nested, direct arguments are replaced by shorter indirect ones, and logically necessary, but easily inferable arguments are left out. Rule applications are performed in a staged process, exploiting regularities of the structure of the proof graph to avoid interferences between rules and to obtain as many reductions as possible.

4 Conclusion

Once we have examined the role of anticipation in major components of ΩMEGA, we have to address the question raised by the title of this paper, that is, to what extent anticipation is the driving force in promoting an interactive proof development environment into a mathematical assistant. On the one hand, the techniques outlined here demonstrate the diversity of situations and the capabilities to act adequately therein, in which anticipation appears as a central concept. This has been taken into account in the ΩMEGA system.

On the other hand, for a system to be truly called a mathematical assistant, many of these techniques need to be elaborated in greater depth and sophistication and complemented by some other techniques such as knowledge representation and reasoning skills of various sorts, in which anticipation plays a subordinate role if at all. Hence, we might conclude that anticipation is without doubt a central concept needed for building a mathematical assistant, but its adequate interpretation and the elaboration of its manifestations require considerable effort.

At the moment we are incorporating further techniques in which anticipation manifests itself into the ΩMEGA system. They include the exploitation of structured mathematical knowledge, extensions to the planner so that it can run in a reactive mode, and provision of more natural and convenient ways to specify a problem.

In an extension of anticipation in proof presentation, we also investigate the impact of larger sequences of proof lines on the preferred presentation level, which in a number of cases should lead to proof sketches or partial presentations rather than to complete proof documentations (Horacek, 1999). Altogether, the presentation subsystem attempts to anticipate a number of communication problems which may arise due to insufficiently adapted presentation contents, and the system applies techniques that take into account the impact of users' capabilities for tailoring these presentations.

References

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 Conference on Automated Deduction* (pp. 252–255). Townsville, Australia: Springer Verlag.

Benzmüller, C., & Sorge, V. (1998). A blackboard architecture for guiding interactive proofs. In F. Giunchiglia (Ed.), *Artificial intelligence: Methodology, systems and applications* (pp. 102–114). Sozopol, Bulgaria: Springer Verlag.

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

Cheikhrouhou, L., & Siekmann, J. (1998). Planning diagonalization proofs. In J. Denzinger, M. Kohlhase, & B. Spencer (Eds.), *Cade-15 workshop "using ai methods in deduction"* (pp. 3–12).

Horacek, H. (1998). Generating inference-rich discourse through revisions of RST-trees. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence AAAI-98 and Tenth Conference on Innovative Application of Artificial Intelligence IAAI-98* (pp. 814–820). MIT Press.

Horacek, H. (1999). *Tailoring inference-rich descriptions through making compromises between conflicting cooperation principles.* (Under review for *International Journal of Human Computer Studies*)

Huang, X., & Fiedler, A. (1997). Proof verbalization as an application of NLG. In M. E. Pollack (Ed.), *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI).* Nagoya, Japan: Morgan Kaufmann.

Kerber, M., Kohlhase, M., & Sorge, V. (1998). Integrating computer algebra into proof planning. *Journal of Automated Reasoning.* (Special Issue on the Integration of Computer Algebra and Automated Deduction; forthcoming)

Melis, E. (1990). Ai-techniquese in proof planning. In H. Prade (Ed.), *Proceedings of of the 13th european conference on artifical intelligence* (pp. 494–498). Brighton, UK: John Wiley & Sons, Chichester, UK.

Melis, E. (1998). The Heine-Borel challenge problem. in honor of Woody Bledsoe. *Journal of Automated Reasoning*, *20*, 255–282.