From MKRP to ΩMEGA

Manfred Kerber

The University of Birmingham,
School of Computer Science,
Birmingham, B15 2TT, England
M.Kerber@cs.bham.ac.uk
http://www.cs.bham.ac.uk/~mmk

Abstract. Around 1990 the work on the first-order theorem prover MKRP stopped after a development going on for more than a decade. Instead a new system has been developed since then, the mathematical assistant ΩMEGA. In this contribution I try to summarise some of the discussions and decisions that led to this shift in focus and to the development of the ΩMEGA system, and I attempt in retrospect to give a tentative evaluation of some of the decisions.

1 Introduction

In the late 1980's the MKRP-project came to an end, after a development of more than ten years in which a couple of millions of Deutschmark were spent (a Deutschmark is roughly half a Euro or half a US-Dollar). A new project on a new system started, the ΩMEGA system[1]. In the preceding discussions many decisions were taken and paradigms discussed. Now more than 10 years later it may be worthwhile firstly to document some of the discussions – also since the whole endeavour was a significant step in Jörg Siekmann's work in Mechanised Theorem Proving, and secondly to try a cautious first evaluation of some of the decisions.

The MKRP-project started in Karlsruhe when Jörg Siekmann was a research fellow at the University of Karlsruhe and was continued when he took up there a professorship for artificial intelligence at the University of Kaiserslautern. In this paper some details about MKRP and the MKRP-project are discussed, but only insofar as they are relevant for the transition from MKRP to ΩMEGA. It is not a description of the development of MKRP (neither of ΩMEGA), but a report on the transition period.

In order to understand the transition let's first take a look at the context in which the decisions were made, then take a closer look at the MKRP system itself and discuss shortcomings of the system in the intended applications and means to overcome these.

[1] Initially the new system was called Ω-MKRP, later only Ω or Omega and at a time the logo ΩMEGA was introduced.

2 The Context

Preceding the birth of ΩMEGA a major shift in the work of Jörg Siekmann and his group as well as their working conditions took place. Jörg was at that time an (associate) professor (C3) for artificial intelligence at the Computer Science department of the University of Kaiserslautern and headed a group of around ten active teaching and research assistants mainly financed by the German National Science Foundation (Deutsche Forschungsgemeinschaft, DFG), mainly, in the Sonderforschungsbereich 314 on Artificial Intelligence and European Union Esprit projects. Only the DFG financed projects were directly related to MKRP. In Germany, research assistants and teaching assistants typically work towards their PhD alongside their project work, only the professor has a permanent position. At the end of the 1980's most researchers in Jörg Siekmann's group were in the final stages of their PhD theses. Some had already left the group, some were about to leave, or looking for different activities which went beyond MKRP. To a large degree the work in the final phase of the MKRP-project and the potential extensions can be described by the work done by the people working on it. I will briefly mention some of these results of people working in the core of MKRP.

MKRP was built on the so-called clause graph procedure, originally proposed by Robert Kowalski [Kow75]. In this approach, a graph is used to store all possible resolution steps, detect redundancies and simplification possibilities. A difficult question is whether the procedure is actually complete and confluent, that is, whether if one starts with an unsatisfiable clause set, from any intermediate state it is still possible to derive the empty clause, and – assumed a fair strategy is employed – the empty clause will eventually be derived.

Norbert Eisinger [Eis91] studied the theoretical properties of clause graph procedure like completeness and confluence. Already during the Karlsruhe phase, Christoph Walther [Wal83] developed the order-sorted logic[2] on which MKRP is based. In Kaiserslautern, Manfred Schmidt-Schauß [SS89] worked on extensions of this logic with term declarations. Hans-Jürgen Bürckert [Bür90] worked on constraint resolution, Alexander Herold [Her87] on the combination of unification algorithms, and Christoph Lingenfelder [Lin90] on the presentation of resolution proof in natural deduction. Karl Hans Bläsius [Blä86] and Axel Präcklein [Prä92] worked on different approaches to equality reasoning from human-oriented to traditional rewrite oriented ones. Hans Jürgen Ohlbach (but also many others) were very involved in the MKRP project or related projects. They played a crucial rôle in the development of MKRP and of extensions to MKRP. Hans Jürgen in the end chose a PhD topic on modal logic rather than MKRP-related issues [Ohl88].

While all these people finished their PhDs, two things crucial for the further development happened around the same time, firstly the German Research Centre for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) was founded in Saarbrücken and Kaiserslautern, and secondly Jörg Siekmann was offered and accepted a full professorship (C4) in Saarbrücken, jointly with the post of a director at the DFKI. This meant not only that a shift

[2] The idea of sorts is to replace particular unary predicate symbols by sort symbols. This has the advantage to shorten clauses and prevent meaningless unifications.

in the focus of his work was necessary, but also that senior people were needed to head the new research areas in the DFKI.

Initially, the ΩMEGA-group (in addition to Jörg Siekmann himself) consisted of Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Erica Melis, Dan Nesmith, and Jörn Richts. While Xiaorong and Manfred had worked in the group for a couple of years already, Xiaorong on the verbalisation of natural deduction proofs, and Manfred on human-oriented theorem proving, the others recently joined. Michael had just met Jörg Siekmann at a seminar of the prestigious Studienstiftung. His special interest was to develop a logic close to the mathematical representation found in mathematical practice, concretely this meant to develop a sorted higher-order logic and its mechanisation. Erica had worked on analogy and related questions in a general context at the Humboldt University in Berlin; she joined the group after she met Manfred just before the fall of the wall at a conference on analogy in East Germany. Dan came – a short while after the others – from Peter Andrews' group to Saarbrücken. He brought in his tremendous experience in the implementation of the TPS system. This way TPS strongly influenced the style of the logical core of ΩMEGA. Dan was to become the main developer of the Keim implementation system, which has provided the implementational base for ΩMEGA [HKK+94]. Jörn had just finished his master thesis on MKRP and had a general interest in the new paradigm of ΩMEGA.

3 Paradigms of the MKRP System

As mentioned MKRP [MGR84,EO86,OS91,Prä92a] is a traditional theorem prover based on resolution and paramodulation for sorted first-order logic. In practice, the MKRP-system is a traditional first-order theorem prover; in spirit, however, Jörg Siekmann had from the very start of the project in mind to build a theorem prover following a human-oriented approach of reasoning. The original idea was to have a so-called logic engine, that is, the graph-based resolution engine, which is guided by a so-called "supervisor," that is, a module which tells the logic engine what to do next (see Fig. 1). Jörg Siekmann's idea was that the logic engine should be supplemented by a human-oriented component that guides the logic engine. However, more and more effort was put into the logic engine itself and the "supervisor" was never realised nor was there a serious attempt to design it. Many people started to work on it, but switched to work on the core system since there was concrete work to be done, which also was easier to sell to a community that is up to now not very interested in human-oriented theorem proving. At a time Axel Präcklein built an interactive component as an extension to MKRP which visualised all possible resolution steps and allowed to run MKRP in an interactive mode. Using this component, Axel Präcklein and Manfred Kerber tried to find patterns in the clause graph which allow to select the next resolution possibility heuristically. However, it turned out that it was already very difficult to find proofs at all this way and virtually impossible to detect any usable structure. Usually MKRP performed these steps much better than humans. From this it was concluded that automating a human-oriented selection module for resolution steps would be very difficult (it wouldn't have been very human-oriented anyway).

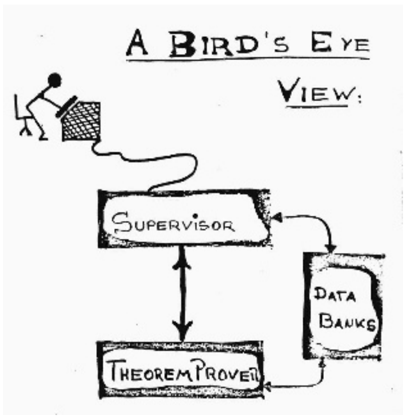

Fig. 1. A bird's eye view for the intended interplay between MKRP and the control component the "supervisor", taken from Jörg Siekmann's lecture notes.

The motivation force in the MKRP project was to prove a mathematical textbook fully using MKRP. The only ever fully proved mathematical textbook is Edmund Landau's "Grundlagen der Analysis" [Lan30]. This book was interactively proved in the Automath system see [NGdV94] by L.S. van Benthem-Jutting [Jut79]. It was a major attempt to prove a full mathematical textbook and it took van Benthem-Jutting five years to do it, although Landau's book is very formalised with 301 theorems on 134 pages. The effort needed indicates that Automath seems not to have been the right tool to prove textbooks. While it is a proof checker, MKRP is an automated theorem prover. The hope was that this would make it much easier to prove a book with MKRP. The textbook chosen was Peter Deussen's "Halbgruppen und Automaten" (Semi-groups and Automata) [Deu71]. The first 5 of 17 sections were actually proved with MKRP. As will be seen in the following, there were, however, severe problems with using MKRP for this task. When we formed plans for a new project, ΩMEGA, we naturally wanted to take into account all the lessons learned from the attempt to prove the text book. In the following the most important insights will be summarised which strongly influenced the further development (see also [Ker92] and [HKK$^+$92]).

Logic. The representation of the mathematical concepts in the sorted first-order input language of MKRP is often clumsy and unnatural, also the representation was often *ad hoc.* The concepts and constructs of a typical mathematics textbook are quite rich and much better approximated by a higher-order language, we were forced to use sophisticated encoding techniques to translate them manually into the MKRP first-order input language. While the availability of sorts and the built-in equality predicate allow for a tolerably adequate translation, it is not always obvious what the theorems proved by MKRP have to do with the textbook theorems and hence what is actually proven. As a minimal requirement one would want an automatic translation technique from higher-order to first-order logic to support a user in the encoding task.

Knowledge Base. The MKRP-system, as most other automated theorem provers, has no integrated *mathematical* knowledge. Each time definitions and lemmas, which are used as preconditions for the actual theorem, must be coded and re-input. This is not only rather boring, but is also a serious source of error. Often (slightly) different formulations are chosen in different contexts, with the consequence that the correctness of the whole procedure of machine verification of textbooks is no longer assured. Moreover, the user may insert lemmas that cannot be proven in the given context. Discipline may be helpful, but as practice shows, automated assistance is indispensable. In short, a system that supports human mathematicians in proving theorems must include a database of mathematical knowledge that can be accessed and updated in a controlled way. This in itself is a major research task, still not fully solved.

Structuring in Subproblems. More often than not, real mathematical theorems are too hard to be proven automatically. This state of affairs can be ameliorated by strengthening the deductive power of the prover in various ways (and since then considerable progress has been made in the field, of course). For every system, however, there exist theorems that cannot be shown automatically. In order to be useful, the user must be given the opportunity to guide the proof process interactively. In a classical theorem-proving system this is almost impossible: the cycle of interaction consists of a complete restart with a different setting of the parameters or a reformulation of the clauses. The main influence the user has, consists in the appropriate choice and formulation of the problem. The way the preconditions of a theorem are selected, for instance, is of paramount importance for the performance of the system. An additional necessary facility is one for splitting the problem manually into subproblems, so that they can be proved separately and then used as lemmas later in the proof of the original theorem. Traditional theorem provers lack such support and the situation is far from satisfactory, as all structuring decisions and all proof plans are hand-crafted. In short, all of this requires too much care and skill from the user, and not surprisingly there are fewer than a handful of well-known experts who are renowned for their skill in proving difficult theorems with the help of a machine. Since it is always possible to break down difficult theorems into digestible subproblems, there is also the question to which degree this procedure can be called automatic.

All these points showed us that we needed a *new* system which should be developed from scratch and that there was no point in attempting to extend the existing MKRP system. However, we were also very reluctant to throw away more than ten years of development work and dozens of person years' work. This led to the idea to build an open system, in which it is possible to add external systems to solve subproblems. The first external system would be MKRP, others could then easily follow.

4 Discussions and Decisions

The principles for the ΩMEGA system were developed in many meetings, seminars and discussions, a picture of one of the blackboard summaries can be found in Fig. 2. These discussions were influenced by many people, who either came to the group to give presentations or with whom we were in intensive discussions. To mention just a few: Jörg Denzinger (with whom we had intensive discussions on proof planning), Christoph Kreitz (who explained to us the details of Nuprl), Wolfgang Reif (on the reuse of proof attempts), William Farmer (on the IMPS system). In addition, we had numerous discussions with Dieter Hutter, Claus Sengler, Jürgen Cleve and others, who just arrived in Saarbrücken at the same time as we, to work on Jörg Siekmann's verification project. Furthermore we read many articles outside the main paradigm of the MKRP system, such as papers on Automath [Bru80,NGdV94], Isabelle [Pau90], and proof planning [Bun88].

These discussions were about almost all aspects of a system, from the high-level philosophy of the system to the implementation language and the data structures employed. In the remainder of this section I want to summarise some key questions and decisions. Let us first take a look at the philosophy of the system. In this we took up the experience gathered in the work of proving the first third of [Deu71].

4.1 Automated Versus Interactive

Certainly a mathematician wants to guide the process of proving mathematical theorems. A system that puts you before an "all or nothing" approach seems highly unsatisfactory. Ideally you want a system that is as automatic as possible in order to support a user, but leave as much room for interaction as possible at the same time.

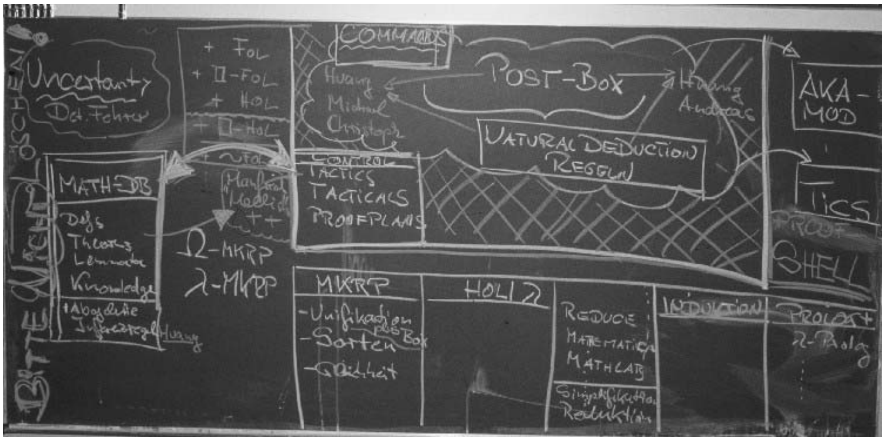

Fig. 2. A Blackboard Summary of Discussions in 1991 written by Jörg Siekmann.

A quote from [HKK$^+$94a]:

On account of this, we believe that significantly more support for proof development can be provided by a system with the following two features:

– *The system must provide a comfortable human-oriented problem-solving environment. In particular, a human user should be able to specify the problem to be solved in a natural way and communicate on proof search strategies with the system at an appropriate level.*
– *Such a system is interesting only if it relieves the user of non-trivial reasoning tasks and provides the foundation for a practicable increased reasoning power. We are convinced that this requires not only task-specific tactics but also the strong reasoning power of a general logic engine.*

As a consequence we decided to build a system that on the one hand allows for tactical theorem proving, which allows to influence the proof search by calling tactics, and on the other hand has integrated strong external components like automated theorem provers which can be called as black boxes to solve subproblems (see Fig. 3). In summary we tried to find a viable compromise between automatic and interactive theorem proving. In retrospect, the decision to follow an automatic and interactive approach at the same time seems to have been the right one. The full strength of this synthesis seems to be coming to the fore only recently in work on agent-oriented theorem proving as developed by Christoph Benzmüller and Volker Sorge [BS01] and multi initiative proof planning by Andreas Meier and Erica Melis [MM00].

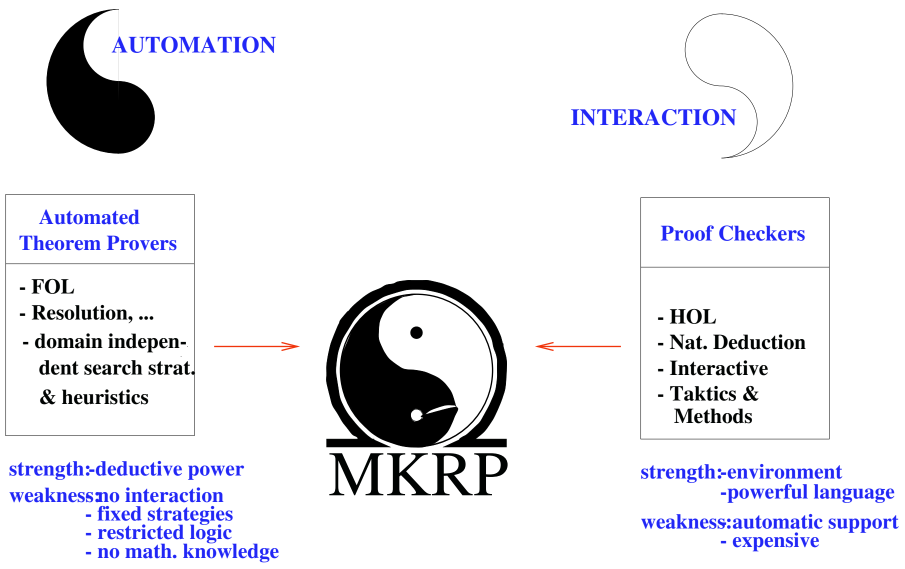

Fig. 3. Motivation Diagram for the ΩMEGA system.

4.2 Paradigm

Related to (but independent from) the question whether to follow an automated or an interactive theorem proving style, was the question what the type of the system should be, should it follow traditional machine-oriented theorem proving or human-oriented theorem proving.

Here again, we followed the approach that we wanted to include different paradigms. We wanted to include theorem provers like MKRP and Otter, but wanted also to take the original idea of a human-oriented approach very seriously (see Fig. 3). Some work on analogical theorem proving [Ker89] was done at that time. Around the same time we understood the significance of Alan Bundy's approach to use planning techniques in theorem proving in the form of proof planning [Bun88]. It was at the same time when the last proposal to reimplement MKRP and to bring it up to the most recent developments in resolution style theorem proving was discussed. We decided against this and started concrete work on the ΩMEGA system instead. We decided to make use of fast existing first-order theorem provers like Otter rather than to bring MKRP to speed.

The approach of proof planning looked very attractive since it allows – unlike MKRP – to include domain-dependent reasoning knowledge in form of proof planning methods [MS99]. Around this time we also adopted the idea that for automation incomplete approaches can be stronger than complete ones.

The overall architecture of ΩMEGA as discussed then and realised later can be seen in Fig. 4. Centred around the structure of Natural Deduction proofs, different components and the user can manipulate partial proofs to complete a proof, this can be done by inserting information from a knowledge base, by proof planning, by external components, or proof transformation.

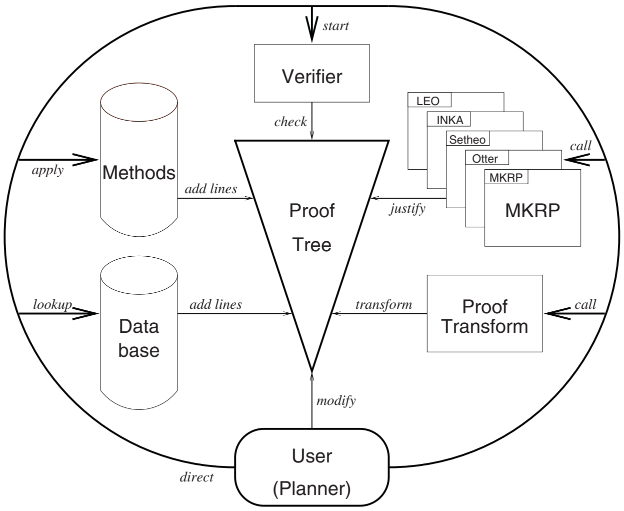

Fig. 4. Architecture of ΩMEGA.

4.3 The Logic Language – First-Order Versus Higher-Order

MKRP is a first-order theorem prover. For all theorems we tried to prove we found a representation in first-order logic. Also there has been work by Robert Boyer et al. [BLM$^+$86] (later taken up by Art Quaife [Qua92]), which shows how first-order logic can be effectively used to prove large chunks of mathematics in some form of set theory. This is also the approach taken in Mizar [Rud92]. The advantage is that a few axioms suffice to build up all of the relevant mathematics. The disadvantage is that the approach seems to be far away from mathematical practice and a human-oriented approach. When you want to represent a function in mathematics, a function symbol in logic seems to be much closer to the informal notion than a left total and right unique relation.

The argument was strongly supported by the plan to build on existing sort systems (like that used in MKRP) to extend it to higher-order logic, so that it would be possible to speak about unary functions on real numbers, continuous, and differentiable ones and so on. Here two different aspects had to be considered. Firstly we wanted to use the language as the representation language, that is, we wanted it to be as strong as possible (including dependent sorts, to be able to encode structures like groups very naturally as a set $G : set$ with an operation $+ : G \times G \to G$, where the operation depends on G). Secondly Michael Kohlhase wanted to (and later did in collaboration with Christoph Benzmüller) build a theorem prover based on higher-order logic with sorts which should form the Logic Engine for Omega, Leo [Koh94,BK98]. Since this is a very difficult task, dependent sorts (as well as partiality as discussed in [KK96]) were not included for the time being.

4.4 Explicit Proofs, Proof Presentation, and Interface

The importance of an adequate presentation of proofs has always played a significant rôle in the MKRP as well as the ΩMEGA project. This started by the work of Christoph Lingenfelder [Lin89,Lin90,LP91] and Xiaorong Huang [Hua96] and has been continued by Armin Fiedler since. From the work to present proofs to humans it was a short step to arrive at the philosophical position that proofs must be checkable [HKK$^+$94c]. This solved a major problem of the structure of the ΩMEGA system, namely, how is it possible to ensure correctness of proofs in the presence of many – quite heterogeneous black box components – which may contribute to a proof.

The ΩMEGA system went first with an Emacs interface only. In the initial discussions we discussed already the importance of a high-level user interface, but in the implementation we had to focus on the kernel of the system. Only later LOUI [SHB$^+$99] was developed to provide a graphical user interface to ΩMEGA.

4.5 Knowledge Representation

The representation of mathematical knowledge was considered of great importance from the very beginning [SK93]. It was considered as a central question and attempted even in days when there were still discussions to build a so-called "supervisor" for MKRP. In discussions between Norbert Eisinger, Jörg Siekmann, and Manfred Kerber it was decided to start the work on a knowledge based reasoning system with the representation of knowledge. It has been considered important to represent mathematics not only as mathematical formulae, but also to attribute to a formula a semantic status, whether it is an assumption, a precondition, a definition, a conjecture, or a theorem. Furthermore a context should be provided, problems can build up on each other, which makes inheritance of concepts possible. A particular interest was also put in re-representation issues [KP96], however, the question is still mainly unanswered.

4.6 External Systems

As external system to include we started with MKRP, also to justify that all the work that went into MKRP was not in vain. This led then to the next step to include other first-order theorem provers, which are much faster than MKRP, like Otter [McC90]. While this approach was criticised by reviewers, since we didn't follow a seamless approach of user-oriented theorem proving which employs human-oriented theorem proving like proof planning throughout, it made it possible to further extend the system later on. For instance, Leo [BK98] was anticipated as a loosely coupled and not tightly integrated system. Computer algebra systems followed [KKS98] as well as constraint solvers [MZM00]. This diversified the type of systems included. A generalisation of such a flexible integration was never anticipated in the early days, but the high flexibility needed can be viewed in retrospect as a seed which helped to generate the ideas of Mathweb [FK99] and ⓂDOC [Koh01] by Michael Kohlhase and others.

4.7 Application

As intended application for the ΩMEGA system we took over the old project of MKRP to prove a mathematical textbook without any discussion. In hindsight this is too narrow a view what such a system can be used for. Although we never explicitly excluded other potential applications, it might have been better to positively keep other potential applications – like the recently emerging applications in education (see Erica Melis' contribution in this volume) – in sight at an early stage already.

4.8 Programming Language

Different programming languages were discussed when we started the ΩMEGA project, to mention some, C (for efficiency), Prolog (for rapid prototyping), Lisp (as a compromise, which stood in the tradition of the MKRP project), and ML (viewed as a typed variant of Lisp). The decision against C or C++ was in retrospect good, since our ideas were not clear enough when we started with ΩMEGA that we could exactly specify it, Prolog may not have been flexible enough, ML with its type system might have been the better choice and some people favoured it when we started, but in the end tradition prevailed and we decided for Lisp, because of the knowledge, the hardware, and the software available. Lisp proved to be a powerful, flexible language which allowed for many developments in the sequel, functional, object-oriented and concurrent system development. It was a good choice (although ML might have been a better).

ΩMEGA was not directly implemented in Lisp (Common-Lisp to be more precise), but in the Keim environment, which was a programming environment built on top of Lisp (in the German Focal Programme on Deduction, financed by the Deutsche Forschungsgemeinschaft). Its philosophy is (quote from [HKK$^+$94]):

> *..., those who wish to apply techniques developed by the theorem-proving community face the choice of either learning this 'black art' themselves by developing their own prover from scratch, or jury-rigging available provers to get some kind of result.*

While Keim greatly facilitated the development of ΩMEGA, it remained still a 'black art' to build and extend ΩMEGA. Keim turned out to be a complex system, which is difficult to handle. Furthermore there are performance problems. In retrospect I think these are due to the attempt to build with Keim a system which is very general and construed to enable the implementation of a wide range of deduction systems. The question whether to build a system as general as possible versus to build a system as simple as possible was answered in favour of generality. This seems to me now to have been a mistake.

5 Conclusion

Most decisions turned out to be fruitful although they might not all have been optimal. As already mentioned an earlier focus on a different application domain and a different approach to the programming might have been beneficial. But by far not all developments were foreseen. In some aspects we were too optimistic what could be achieved in ten years – for instance, we thought that it would be much easier to formalise standard mathematics in sorted higher-order logic. In other aspects we didn't dare to hope for a state like the one achieved today – for instance, we didn't hope that the work would be applicable now already in education.

There are developments we did not anticipate, but which were made possible by ΩMEGA. I want to mention agent-oriented theorem proving [BS01], knowledge-based proof planning [MS99], Mathweb [FK99], and ⓂDoc [Koh01]. In my view this shows that ΩMEGA has been a flourishing project.

ΩMEGA meant a change of direction in Jörg Siekmann's theorem proving group, a change in area, in approach, and in paradigm. This was of course very risky, since it meant a reduced possibility for publications for a significant amount of time and the potential knock-on effect on funding. Fortunately referees in the German funding organisations were very positive and supportive (I just want to mention Wolfgang Bibel and Michael M. Richter here). I think that ΩMEGA has been a scientific success and a worthwhile enterprise. Jörg Siekmann was already an established scientist who could have rested on his laurels and continued with what he always did, when ΩMEGA started. He, however, actively initiated this major change in direction, which considerably deviated from the path originally envisaged in the MKRP project, when he was convinced that only in this way we would come closer to making the old dream true to automate mathematics. Leibniz had a dream, "Calculemus", namely to mechanise mathematics (actually he wanted to mechanise not only mathematics but all of human thought, we never were so ambitious in the ΩMEGA project). There are many problems in detail still to be solved to provide a powerful useful tool for mechanised mathematics. In the past there have been too many promises/predictions about what will be achieved in the near future. I don't want to add another one, but although we have not yet realised the dream, I feel that we are significantly closer to its realisation.

This paper is not a survey on the ΩMEGA system, but just on the discussions during the transition from MKRP to ΩMEGA. Many exciting developments have taken place since then and without doubt will take place in the future. They have to be reported somewhere else.

Acknowledgements

The transition from MKRP to ΩMEGA was possible since several things came together. Not only the researchers involved were filled with great enthusiasm, but also many undergraduate students worked with great enthusiasm on the project.

I feel that the description which I tried to provide here is more incoherent and subjective than I would like it to be; this is entirely my fault. That it is not even more incoherent is due to many helpful comments by Norbert Eisinger, Michael Kohlhase, and Erica Melis on a earlier draft. I would like to express my thanks to them hereby.

References

[BK98] Christoph Benzmüller and Michael Kohlhase. Leo – a higher-order theorem prover. In Claude Kirchner and Hélène Kirchner, eds., *Proceedings of the 15th CADE*, p. 81–97, Lindau, Germany, 1998. Springer, LNAI 1421.

[Blä86] Karl Hans Bläsius. *Equality Reasoning Based on Graphs.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1986.

[BLM$^+$86] Robert Boyer, Ewing Lusk, William McCune, Ross Overbeek, Mark Stickel, and Lawrence Wos. Set theory in first-order logic: Clauses for Gödel's axioms. *Journal of Automated Reasoning*, **2**:287–327, 1986.

[Bru80] Nicolaas Govert de Bruijn. A survey of the project Automath. In J.P. Seldin and J.R. Hindley, eds., *To H.B. Curry - Essays on Combinatory Logic, Lambda Calculus and Formalism*, p. 579–606. Academic Press, London, UK, 1980.

[BS01] Christoph Benzmüller and Volker Sorge. Oants – an open approach at combining interactive and automated theorem proving. In Manfred Kerber and Michael Kohlhase, eds., *Symbolic Calculation and Automated Reasoning: The CALCULEMUS-2000 Symposium*, p. 81–97, St. Andrews, Scotland, 2001. A.K. Peters, USA.

[Bun88] Alan Bundy. The use of explicit plans to guide inductive proofs. In Ewing Lusk and Ross Overbeek, eds., *Proc. of the 9th CADE*, p. 111–120, Argonne, Illinois, USA, 1988. Springer, LNCS 310.

[Bür90] Hans-Jürgen Bürckert. *Constraint Resolution – A Resolution Principle for Clauses with Restricted Quantifiers.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1990.

[Deu71] Peter Deussen. *Halbgruppen und Automaten*, Volume 99 of *Heidelberger Taschenbücher.* Springer, 1971.

[Eis91] Norbert Eisinger. *Completeness, Confluence, and Related Properties of Clause Graph Resolution.* Pitman, London, UK, 1991.

[EO86] Norbert Eisinger and Hans Jürgen Ohlbach. The Markgraf Karl Refutation Procedure (MKRP). In Jörg H. Siekmann, ed., *Proc. of the 8th CADE*, p. 681–682, Oxford, UK, 1986. Springer.

[FK99] Andreas Franke and Michael Kohlhase. System description: MATHWEB, an agent-based communication layer for distributed automated theorem proving. In Harald Ganzinger, ed., *Automated Deduction – CADE-16*, LNAI 1632, p. 217–221. Springer, 1999.

[Her87] Alexander Herold. *Combination of Unification Algorithms in Equational Theories.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1987.

[HKK$^+$92] Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Erica Melis, Dan Nesmith, Jörn Richts, and Jörg Siekmann. Omega-MKRP – a proof development environment. Seki Report SR-92-22, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1992.

[HKK$^+$94] Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Erica Melis, Dan Nesmith, Jörn Richts, and Jörg Siekmann. KEIM: A toolkit for automated deduction. In Alan Bundy, ed., *Automated Deduction – CADE-12*, Proceedings of the 12th International Conference on Automated Deduction, p. 807–810, Nancy, France, 1994. Springer. LNAI 814.

[HKK$^+$94a] Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Erica Melis, Dan Nesmith, Jörn Richts, and Jörg Siekmann. Ω-MKRP: A proof development environment. In Alan Bundy, ed., *Automated Deduction – CADE-12*, Proceedings of the 12th International Conference on Automated Deduction, p. 788–792, Nancy, France, 1994. Springer. LNAI 814.

[HKK$^+$94c] Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Dan Nesmith, and Jörn Richts. Guaranteeing correctness through the communication of checkable proofs (or: Would you really trust an automated reasoning system?). In David Basin, Fausto Giunchiglia, and Matt Kaufmann, eds., *Proceedings of the CADE-Workshop on Metatheoretic Extensibility of Automated Reasoning Systems*, p. 31–33, Nancy, France, 1994.

[Hua96] Xiaorong Huang. *Human Oriented Proof Presentation: A Reconstructive Approach.* infix, St. Augustin, Germany, 1996.

[Jut79] L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the* Automath *System*, Volume 83 of *Mathematical Centre Tracts.* Mathematisch Centrum, Amsterdam, The Netherlands, 1979.

[Ker89] Manfred Kerber. Some aspects of analogy in mathematical reasoning. In Klaus P. Jantke, ed., *Analogical and Inductive Inference; International Workshop AII '89*, p. 231–242, Reinhardsbrunn Castle, GDR, October 1989. Springer. LNAI 397.

[Ker92] Manfred Kerber. *On the Representation of Mathematical Concepts and their Translation into First Order Logic.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1992.

[KK96] Manfred Kerber and Michael Kohlhase. A tableau calculus for partial functions. *Collegium Logicum – Annals of the Kurt-Gödel-Society*, **2**:21–49, 1996.

[KKS98] Manfred Kerber, Michael Kohlhase, and Volker Sorge. Integrating computer algebra into proof planning. *Journal of Automated Reasoning*, 21(3):327–355, 1998.

[Koh94] Michael Kohlhase. *A Mechanization of Sorted Higher-Order Logic Based on the Resolution Principle.* PhD thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1994.

[Koh01] Michael Kohlhase. ⓂDoc: Towards an internet standard for the administration, distribution and teaching of mathematical knowledge. In Eugenio Roanes Lozano, ed., *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2000*, LNAI 1930. Springer, 2001.

[Kow75] Robert Kowalski. A proof procedure using connection graphs. *JACM*, **22**, 1975.

[KP96] Manfred Kerber and Axel Präcklein. Using tactics to reformulate formulae for resolution theorem proving. *Annals of Mathematics and Artificial Intelligence*, 18(2-4):221–241, 1996.

[Lan30] Edmund Landau. *Grundlagen der Analysis.* Akademische Verlagsgesellschaft, Leipzig, Germany, chelsea publishing, 1948 Edition, 1930.

[Lin89] Christoph Lingenfelder. Structuring computer generated proofs. In N.S. Sridharan, ed., *Proc. of the 11th IJCAI*, p. 378–383, Detroit, Michigan, USA, 1989. Morgan Kaufmann, San Mateo, California, USA.

[Lin90] Christoph Lingenfelder. *Transformation and Structuring of Computer Generated Proofs.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1990.

[LP91] Christoph Lingenfelder and Axel Präcklein. Proof transformation with built-in equality predicate. In John Mylopoulos and Ray Reiter, eds., *Proc. of the 12th IJCAI*, p. 165–170, Sydney, 1991. Morgan Kaufmann, San Mateo, California, USA.

[McC90] William McCune. Otter 2.0. In Mark E. Stickel, ed., *Proc. of the 10th CADE*, p. 663–664, Kaiserslautern, Germany, 1990. Springer, LNAI 449.

[MGR84] Karl Mark G Raph. The Markgraf Karl Refutation Procedure. Technical Report Memo-SEKI-MK-84-01, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1984.

[MM00] Erica Melis and Andreas Meier. Proof planning with multiple strategies. In John Lloyd et al., ed., *Proc. of First International Conference on Computational Logic – CL 2000*, p. 644–659, Berlin, Germany, 2000. Springer, LNAI 1861.

[MS99] Erica Melis and Jörg H. Siekmann. Knowledge-based proof planning. *Artificial Intelligence*, **115**(1):64–105, 1999.

[MZM00] E. Melis, J. Zimmer, and T. Müller. Integrating constraint solving into proof planning. In Ch. Ringeissen, ed., *Frontiers of Combining Systems, Third International Workshop, FroCoS'2000*, p. 32–46. Springer, LNAI 1794, 2000.

[NGdV94] Rob Nederpelt, Herman Geuvers, and Roel de Vrijer, eds. *Selected Papers on Automath*, Volume 133 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, The Netherlands, 1994.

[Ohl88] Hans Jürgen Ohlbach. *A Resolution Calculus for Modal Logics*. PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1988.

[OS91] Hans Jürgen Ohlbach and Jörg H. Siekmann. The Markgraf Karl Refutation Procedure. In Jean-Louis Lassez and Gordon Plotkin, eds., *Computational Logic – Essays in Honor of Alan Robinson*, Chapter 2, p. 41–112. MIT Press, Cambridge, Massachusetts, 1991.

[Pau90] Lawrence C. Paulson. Isabelle: The next 700 theorem provers. *Logic and Computer Science*, p. 361–386, 1990.

[Prä92] Axel Präcklein. *Integration of Rewriting, Narrowing, Compilation, and Heuristics for Equality Reasoning in Resolution-Based Theorem Proving.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1992.

[Prä92a] Axel Präcklein, ed. The MKRP User-Manual. SEKI Working Paper SWP-92-03, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1992.

[Qua92] Art Quaife. Automated deduction in von Neumann-Bernays-Gödel set theory. *Journal of Automated Reasoning*, **8**(1):91–146, 1992.

[Rud92] Piotr Rudnicki. An overview of the Mizar project. Bastaad, Sweden, 1992.

[SHB$^+$99] Jörg Siekmann, Stephan Hess, Christoph Benzmüller, Lassaad Cheikhrouhou, Armin Fielder, Helmut Horacek, Michael Kohlhase, Karsten Konrad, Andreas Meier, Erica Melis, Martin Pollet, and Volker Sorge. Loui: Lovely omega user interface. *Formal Aspects of Computing*, **11**:326–342, 1999.

[SK93] Barbara Schütt and Manfred Kerber. A mathematical knowledge base for proving theorems in semigroup and automata theory – Part I. SEKI Working Paper SR-93-02, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1993.

[SS89] Manfred Schmidt-Schauß. *Computational Aspects of an Order-Sorted Logic with Term Declarations*. Springer, LNAI 395, 1989.

[Wal83] Christoph Walther. A many–sorted calculus based on resolution and paramodulation. In *Proc. of the 8th IJCAI*, p. 882–891, Karlsruhe, Germany, 1983. Morgan Kaufmann, San Mateo, California, USA.