SIXTH FRAMEWORK PROGRAMME
Information Society Technologies
Call FP6-2002-IST-C

Contract for:

COORDINATION ACTION
Annex1 – "Description of Work"

Project acronym: TYPES

Project full title: Types for Proofs and Programs

Proposal/Contract no.: 510996

Related to other Contract no.: (to be completed by the Commission)

Date of preparation of Annex 1: August 6, 2007

Start date of contract: Sept 1, 2004

Contents

1 Project summary

Technology for building accurately specified and reliably correct computer systems is of deep interest for industry in Europe. This is a long-term problem, requiring progress in many areas. The use of formal logics for specification and reasoning about systems, and of strongly typed programming languages with inherent correctness properties, are well accepted as necessary parts of the solution.

The aim of our research activities is to develop the technology of formal reasoning and computer programming based on Type Theory, by improving the languages and computerised tools for reasoning, and by applying the technology in several domains such as analysis of programming languages, certified software, formalisation of mathematics and mathematics education.

This proposal is based on the strong collaboration and achievements in four successful European projects (ESPRIT BRA and Working Group), in which we have built several computer systems for proof development and used them in applications. The theory and the associated systems developed by consortium members define the state of the art in type theory and its applications. Some impressive examples using these systems have been achieved. However, it is a long-term project to make this technology suitable for use by engineers, mathematicians and students. The proposed Co-ordination Action is essential to maintain our community so that sites doing related and complementary work can continue to communicate and collaborate fruitfully.

We request funding for three annual meetings to communicate recent work throughout the consortium, six smaller thematic workshops on designated research themes, one summer school, short courses and short visits between sites.

For wide dissemination of our work, the proceedings of the annual meetings will be formally refereed and published, while the thematic workshops, summer school and short courses will be publically available on a project web page.

2 Objectives of the project and state of the art

Europe needs robust software systems. Society has become dependent on computer systems working in an expected way, but programs are often erroneous. Like all big problems the software crisis can only be solved using a multitude of approaches.

We believe that one such approach is to use the computer to develop provably correct computer systems. Our contribution to this approach spans from fundamental research in logic to concrete formal proofs of major importance. This is an area where Europe has built up considerable competence during the last two decades within the TYPES community.

Research associated with previous EU funding for TYPES[1] has led to logics based on various type theories (e.g. Extended Calculus of Constructions, Calculus of Inductive Constructions), proof tools based on these logics (implemented proof assistants such as Coq, LEGO, Alfa, Isabelle and Mizar), and mathematics libraries which are essential for practical use of the proof tools (e.g. the Mizar mathematical library, the Isabelle developments of set theory and higher-order logic, and the Constructive Coq Repository at Nijmegen). Some of this work is mature, widely used, and proven in significant examples and applications.

Computer assisted formal reasoning is also important for the development of mathematics. Our proof tools have been used to develop machine checked proofs of significant mathematical results. We want to use our work to develop systems that will be useful to mathematicians and other scientists in their proof-oriented work, much as computer algebra systems have been useful for symbolic computations. The preparation and checking of mathematical documents will gain convenience and reliability from machine support. Large libraries of mathematical knowledge can be made available for fast searching with machine support. Proofs from libraries can be studied at greater or lesser levels of detail. Such libraries can support cooperative and distributed working on extended projects. These facilities will also be useful in mathematics education.

However, the language of textbooks and journals in mathematics and computer science is much more convenient, flexible and extensible than the language that any logic or implemented proof tool currently supports. To attain "natural mathematical language" with high level automated support in a formal system is a very long term project. Our plan to progress on these issues has the following four parts:

1. Correctness of Computer Systems: tools and techniques aimed specifically at application of formal methods to system correctness, e.g. programming language specific tools and problem-specific automation of proof search.

2. Formal Mathematics and Mathematics Education: this is the prototype example for *proof in the large*, including very high level *mathematical vernacular* languages, the construction and use of necessarily large libraries of previous work, and distributed working on long-term projects.

3. Proof Technology: the details of proof, including unification, resolution, rewriting, general proof search, tactic languages and *declarative* proof languages.

4. Foundational Research: underlying the previous three areas must be research on the expressiveness and relative correctness of the foundational logics, including syntax, semantics, definitional mechanisms, allowed computation and subtyping.

[1] Esprit Action 3245 (Logical Frameworks), Esprit BRA 6453 (Types for Proofs and Programs), Esprit Working Group 21900 (Types for Proofs and Programs), IST Thematic Network 29001 (Computer-Assisted Reasoning Based on Type Theory).

These ideas must be implemented; practical improvement of our working systems is an important part of our objectives. This includes further development of proof assistants, user interfaces and interfaces between tools so they become capable of supporting realistic scale work in many areas.

2.1 Correctness of Computer Systems

Recent work has shown that type-theory based theorem provers can be used practically in the production of zero-fault or low-fault software. Advances in logical foundations and proof technology have made the application of type theory based proof systems accessible to a large community of programmers and software engineers for wide areas of application. Some spectacular examples are:

- The execution platform for the JavaCard language has been completely formalised, and security properties of this platform have been verified formally, as part of the European IST project Verificard.
- A formal description of the IEEE 754 standard for floating point operations has been developed, and used to verify formally the correctness of a library of tools to perform high-precision arithmetic.
- Computer algebra algorithms have been formalized and proved correct. We also include in this area algorithms for the verification of large boolean formulas, which traditionally play a significant role in the formal verification of embedded software.

Objectives

Program extraction from proofs (both constructive and classical). Our consortium is among the world leaders in this area (e.g. the Coq and MINILOG proof systems are fundamentally designed to support program extraction). Recent large proofs in Coq on constructive analysis have provided a serious challenge to the existing Coq technology.

Language specific verification tools on top of general proof tools. The *Why* and *Krakatoa* tools are verification generators for ML, C and Java programs, including imperative aspects. They produce proof obligations for several general proof tools: Coq, PVS and HOL light. This approach will be significantly developed, with interesting worked examples.

Programming with dependent types A very new topic of our consortium is functional programming languages with dependent types, making the type system much more expressive than well-known languages such as ML and Haskell. In this way, the types of programs can express more interesting program properties than the simple types of ML and Haskell. This contributes to writing correct programs in the first place, allows more expressive module interfaces for modular development, and replaces some human proof by compiler typechecking. Many areas of theory and pragmatics need work to make this idea widely usable, including work on recursive definitions and termination, dependent pattern matching, and compilation. This objective certainly involves prototype development.

Practical challenges Progress on foundations, on our proof tool implementations, and on libraries of proofs to build on, will make it possible to address more challenging examples. We believe further work on programming language specifications, and correctness of program development platforms, like the previous work on Java and JavaCard in our consortium, is one area where our project can have impact on end-users.

2.2 Formal Mathematics and Mathematics Education

The focus of this theme is to adapt our proof assistants to be useful to research mathematicians and other users of mathematics, and in mathematics education. The theme also involves developing and using large scale libraries of formal mathematics, and work on mathematics education material based on formal mathematics that students can study and extend with machine support.

Developing, presenting and reading mathematical theories, including definitions, theorems and proofs, are activities at the heart of the research mathematicians work. These activities are also important in other disciplines where mathematics is used, for example in computer science, where rigorous validation of properties of software systems can be established via mathematical proof.

Just as many mathematicians use computer typesetting systems like LaTeX to produce their text and use computer algebra systems for symbolic computation, so we envisage that eventually many mathematicians will want to use computer systems for support in developing, presenting, storing and searching all aspects of mathematical knowledge.

Furthermore, these mathematical activities use highly specialised skills that can be difficult to learn and to teach. Our computer systems should help in mathematics education. Material can be presented at different levels of detail, from informal to completely formal. Student exercises can be supported by an underlying formal proof tool with a library of the theories, definitions and lemmas as previously presented in the course.

The role of type theory The mechanisms of dependent type theory are implicitly used in standard mathematical practise, as was recognised long ago in de Bruijn's Automath project. Similarly, most computer algebra systems use dependent typing, albeit often implicitly. These are among the many reasons that our proposal is based around the use of dependent type theory.

Objectives

Mathematics repository A large repository (library) of formalised mathematical results is needed for a proof assistant to be useful for non trivial tasks. In fact, libraries must be developed for specialised domain areas, built on top of more general libraries; e.g. a library for topology based on one for set theory, which itself depends on a library of logic. Such libraries must be managed and used on the web (e.g. using XML technology) to allow cooperation on large scale projects. It is a primary objective of the proposal to develop technology for such libraries, and to develop some libraries themselves. Our consortium contains the world's largest formal mathematics repository (Mizar), and members working particularly on repository technology.

Mathematical vernacular and proof language There is a barrier to the use our systems. On one hand users are accustomed to read and write mathematics in a flexible informal style that allows for gaps in proofs that can be accepted by the sufficiently expert; on the other hand the mathematical language that can be handled by our computer systems is inflexible and formal, requiring much precise detail from the user in order to make machine checking possible. Thus it is difficult both to produce machine checked mathematics and to read it.

Our objective is to develop a mathematical vernacular language to overcome this serious problem. There are several related aspects to be addressed.

- The language used for constructing proofs, which is generally based on some operation of refinement, must be made more *declarative*, as in, e.g., the system Mizar and the Isar dialect of Isabelle.

- It is important to be able to more freely mix top-down and bottom-up proof styles.
- Support for naturally mixing computation and proof.
- Powerful and convenient notational mechanisms are required: dynamically configurable parsing, pretty printing, use of symbols, etc.
- Semi-formal mathematics should be supported, i.e. mathematics that is formal except that suitably documented gaps are allowed.
- Support for higher level mathematical constructions, such as local working environments and modular construction of theories with inheritance from precursor theories.

Many of these points depend on advances in proof technology (section 2.3), but putting them together in a vernacular that will be used by practicing mathematicians is an objective on its own.

Mathematics education Our proof assistants are already used in undergraduate education as tools for teaching elementary logic and discrete mathematics. Our current objective is to take this approach much further, into courses in basic computer science and mathematics. In our experience, many students appreciate the possibility to study exact formal presentations of the material being taught, and even more so, to get detailed feedback on their exercises from a machine proof checker.

Further, this is a way to familiarize a wider public with our tools and techniques, including students, who will become the scientists of the future, and mathematics lecturers. This supports the dissemination of formal methods for future industrial use.

Case Studies We aim to develop case studies in the (semi-)formalisation of areas of advanced mathematics and the use of packages for undergraduate education.

Nijmegen, cooperating with Bologna and other sites in the IST Mowgli project, has done some experiments in creating a web-based interactive mathematical text-book, where the formal mathematics is derived from a Coq repository (C-CoRN, the Constructive Coq Repository at Nijmegen). This has been done by taking IDA (an Interactive course on Algebra, implemented at the Technical University of Eindhoven) and linking it to formal mathematical statements and proofs in C-CoRN. Nijmegen will continue this work.

2.3 Proof Technology

In recent years there has been much improvement in the size and complexity of theories and proofs that can be formalised. For example, our consortium has formally developed proofs of the fundamental theorems of algebra and calculus, of correctness of Buchberger's Algorithm, and a nearly complete formal proof of the four-color theorem. We also have large scale computer science applications, such as a formalisation of Java, JVM, and correctness of a bytecode compiler.

However, in order for use of formal proof tools to become ubiquitous in industry and science, we still must have a radical scaling up of proof tool capabilities. We identify particular technical approaches that we propose to work on.

Computation in proofs: reflection and compilation Clearly computation is part of proof, and one reason for our interest in Type Theory is that it contains a programming language in the logic. In recent years there has been much work on *reflection* of decidable properties as object-level computation, so checking these properties appears as single steps of computation in a proof.

For this to be practical the internal programs must be executed efficiently. Thus proof checking technology must integrate techniques from programming language technology, in particular compilation techniques. Progress in this area has already made contributions to large proofs, notabaly the Coq proof of the four-color theorem. However, compared with programming language compilation, there is much to be done.

Computation in proofs: rewriting The internal programming language of Type Theory is descended from pure logical roots, and has not been very convenient for practical programming. E.g. recursion over data is largely restricted to be structural. In section 2.4 we discuss foundational work towards more flexible function definition, including rewriting, dependent pattern matching, and termination of general recursive definitions. All these techniques need to be implemented and tested in practice, which is just now becoming possible with our current theoretical and practical knowledge. This one area promises to be very significant in scaling up formal proofs in the near future.

Distributed and cooperative working For large problems, even informal mathematics proceeds cooperatively, and for pragmatic use of formal tools cooperation is essential. This possibility has been favored by the development of the web, and of databases accessible through the web. It is fair to say that we are evolving from a system-oriented paradigm to a data-oriented paradigm. Several eXchange Proof Languages have emerged (usually in XML) and we predict that in the future, systems will adapt to be able to process proofs expressed in such universal languages.

Such an open paradigm should favor interoperability of systems. Already tools developed as modules of one system have been adapted to be reused in other systems. For example the generic user interface *Proof General*, and the *Why* tool for verifying imperative programs, that produces goals that can be proved in various proof processing systems. Monolithic systems will probably be cut into pieces in the future and such pieces will be the building blocks of proof environments, exactly like editors, compilers, debuggers, . . . , are building blocks of programming environments.

Interface between proof tools Such an open paradigm should also modify the way proof systems interact with other systems, such as rewriting engines, automated theorem proving systems, model checkers, computer algebra systems, ... The interaction of proof processing systems becomes a major issue when large proofs are concerned. We have several small examples of such interfacing in our previous work, such as an automated first order prover, and a model checker, that produce proofs to be checked by one of our type theory checkers. In this way one gets the advantage of highly technical search with the reliability of foundational proof checking. Much more work in this area is possible. Interaction between proof tools and computer algebra systems is perhaps the most exciting new possibility.

Tactic languages In some traditional proof tools ("LCF style" tools) users can write new proof search programs, *tactics*, that are logically safe, using some kind of abstract interface feature of the programming language in which the proof tool is coded. However, in non-monolithic and distributed proof environments, supporting general XML based vernacular languages, it becomes difficult for the user to know enough about the inner workings of the system to do this. What is needed is high level tactic languages. This is an area that has not received much attention.

Declarative proof In most existing proof tools, proofs are given to the system *operationally*: "first try induction, then simplify every subgoal with as much rewriting as possible, ..." Such proofs can only be written interactively, since the user cannot easily predict what the result of each operation will be. Worse, thay cannot be read at all, as the proof commands say nothing about the current state of the proof let alone about the ideas behind the proof. A few systems (e.g. Mizar and the Isar language of Isabelle in our consortium) support a much more natural, *declarative* style. Much more can be done along these lines. The use of proof technology such as powerful unification, rewriting, and automated search algorithms is important in this area, because in writing and reading informal mathematics, human mathematicians make many simple steps of logic and computation implicitly in their heads. Progress in this area will have a positive impact on the higher level mathematical vernacular language discussed in section 2.2.

Objectives: In summary our main aims under this heading are the following.

- The integration of techniques from programming language technology into our proof processing systems.
- The development of our systems to allow the cooperative distributed creation of large proofs.
- The adaptation of successful parts of proof systems so as to be reusable as modules of other systems.
- The development generic interfaces between different mathematical tools, including autometed provers and computer algebra systems.
- The development of more extensive and useful tactic languages.
- The development of declarative proof interfaces.

2.4 Foundational Research: Expressiveness of logic

Many parts of our workplan involve foundational research. The TYPES community has been particularly effective in making progress on foundational research relevent to the development and use of our proof systems. We aim to continue to do this by publishing our research ideas and, where possible, exploiting them in the design and use of our systems.

Here we mention some of the topics that are clearly needed to support formalization of natural reasoning in mathematics and computer science. The main goal of these topics is to improve the *expressiveness* of formal logics, bringing them closer to informal reasoning.

Inductive definitions Definitional schemas are key to mathematical expressiveness. The *generalised inductive definitions* supported by most of our proof tools do not cover some naturally occurring examples. Recent work on *inductive-recursive definitions* shows one approach to extended definitional schemas. Meta-theoretical study is needed to understand and justify this, and other, new definitional principles.

Extensionality Many type theoretic logics use *intensional* equality. The main practical reason for this choice is to have decidable typechecking, since typechecking plays the role of proofchecking in these systems. However, informal mathematics is extensional, and extensional techniques such as forming quotients would be very convenient for formalization. There has been work on interpreting extensional equality in intensional type theory, however the problem is deep and unresolved. Recent work on inductive definitions, and on programming in dependently typed systems, add interest to this topic.

Language features for structuring large proof developments Serious use of formal methods in computer science or mathematics requires building large libraries of definitions and theorems to support an expanding base of formal knowledge available to all users. This in turn requires support of modular development with abstract interfaces hiding internal details. We follow the basic approach to modularity from functional programming languages, but the issues are more involved for logical systems. Several of our proof tools (Coq and Alfa) now have experimental module systems. Recent work in our group has pointed the way to semantics of first class dependently typed modules, but this needs development.

Subtyping Effective use of modularity requires a notion of *subtyping*: every group is a monoid, and the formal language should reflect this fact. Recent work on the proof theory and semantics of subtyping has made much progress. This is a technically hard area, as has been shown by long term work in the related area of functional programming languages.

Rewriting Computation is a big part of mathematical proof, and type theory, by internalising computation, offers advantages over other formal logics. However, to maintain logical consistency, type theories have depended on definitional schemas that restrict the functions that can be defined and their intensional computational behaviour. Much progress has been made on schemas for rewriting that guarantee logical consistency while supporting greater automation in constructing shorter and more natural proofs. This topic, while requiring more theoretical development to extend the scope of rewriting, promises large benefits for formal proof in the relatively short term.

Termination The same goal addressed by rewriting (above) is being addressed independently by recent work on allowing general recursion for definitions, when suitably restricted to guarantee termination. Recent work suggests that types are a particularly suitable framework for termination checking,

Logical Frameworks The use of proof-theoretically weak but expressive logics as *logical frameworks* in which many other *object logics* can be represented has been a theme through all of the TYPES projects. Theory and implementation of a single framework can thus yield proof tools for many object logics. Also, tools for parsing, proof search, libraries, etc., may work at the framework level, and be applicable generically to represented object logics. Significant progress has recently been made in several aspects of this field.

Abstract syntax and binding One main feature of logical frameworks is the ability to deal with binding syntax, alpha conversion, substitution, and related issues generically. However a long-standing problem has been to justify structural induction over such binding syntax. This is an active topic, both in the TYPES community and among our American collaborators.

Weaker frameworks The idea is for frameworks to be weak, so that their meta theory can easily be developed. It is also important to be able to represent object logics precisely. The goal of representing syntax and proofs as precisely as possible using frameworks with the fewest foundational concepts is of great theoretical interset, and also holds out pragmatic advantages, such as allowing stronger definitional principles on top of such frameworks.

Type Isomorphism This topic has been developed for programming languages, but is new for dependent types. The idea is that types that are not equal, but *isomorphic* in some sense, may be treated as equal for some purposes such as automatic proof search or looking up relevant lemmas in a big library. This is applicable to the essential problem of searching mathematical repositories.

3 List of Participants

Partic. role	Partic. no	Participant name	Participant abbrev.	Country	Enter in month	Exit in month
CO	1	Chalmers University	Chalmers	SE	1	44
CR	2	CNRS/Université Paris 7	CNRS	F	1	44
CR	3	INRIA	INRIA	F	1	44
CR	4	Université Paris Sud	Orsay	F	1	44
CR	5	LMU München	LMU	D	1	44
CR	6	TU München	TU	D	1	44
CR	7	Nijmegen University	Nijmegen	NL	1	44
CR	8	University of Białystok	Białystok	PL	1	44
CR	9	University of Durham	Durham	GB	1	2
CR	10	University of Edinburgh	Edinburgh	GB	1	44
CR	11	University of Manchester	Manchester	GB	1	44
CR	12	University of Torino	Torino	I	1	44
CR	13	University of Udine	Udine	I	1	44
CR	14	University of Warsaw	Warsaw	PL	1	44
CR	15	IoC Tallinn	Tallinn	EST	1	44
CR	16	Royal Holloway, Univ of London	Holloway	GB	3	44

4 Relevance to the objectives of FET Open

High quality longer term research with sound objectives A key theme of FET Open is to support high quality long term research of a foundational nature. One of the purposes of the FET support actions is to support structuring and consolidation of communities doing innovative and long-term research. The current proposal fits these objectives well.

The TYPES consortium consists of high-quality research teams in Europe in the area of computer-assisted formal reasoning based on type theory. The research activities of the TYPES community have been continuously supported by several EU projects (Esprit Action 3245 (Logical Frameworks), Esprit BRA 6453 (Types for Proofs and Programs), Esprit Working Group 21900 (Types for Proofs and Programs), IST Thematic Network 29001 (Computer-Assisted Reasoning Based on Type Theory)) as well as many national projects. Our track record has shown excellent achievements both in research and in collaboration. Our software tools are currently in widespread use by researchers throughout the world. The theoretical developments and practical software tools produced by the TYPES consortium not only represent the state-of-the-art in the field, but have great potential in various applications both in industry and in academic institutions. From the past achievements (modelling the JavaCard architecture, formalised proof of the Fundamental Theorem of Algebra), it is realistic to think that, in the future, critical software certified using proof assistants will become a high-quality standard and that libraries of formal mathematics will be of general use in mathematical education and research. In the long term, most mathematics for engineering, research and education, will be done with machine support (just as most writing is now done with machine support). We are working on many aspects of this long term goal, e.g. syntax, semantics, problems of scale, library searching, library maintenance, user interfaces and lower level enabling technologies such as unification and rewriting.

On the other hand, the long term goal of reliable and secure software built in an economic way is clearly far from solved, and requires continuing research and development. Thus, the consolidation of our consortium is extremely important in realising the promising potentials demonstrated by its previous work. In particular, there is need for persistent and long term effort in order to realise industrial take-up of the techniques and software tools developed so far.

The Coordinated Actions (CA) in the FET Open Scheme support the networking and coordination of research and innovation activities as proposed. The requested support of this proposal (annual meetings, special-theme workshops, summer school, short visits, etc.) will ensure effective coordination of the research activities, supported by the national programmes, and fruitful collaborations of the researchers in the area.

Bold ideas that would involve high risks Several of our members have recently started to investigate the idea of developing a dependently typed programming language. It has always been of theoretical interest that dependent type theory contains a programming language, but now we are starting to address the pragmatic implementation of this fact. There are already UK and Swedish national funded projects along these lines: at Royal Holloway *Epigram: Innovative Programming via Inductive Families*, Principal Investigator Zhaohui Luo, funded by UK Research Council EPSRC (GR/R72259); at Chalmers *From Proof Systems to Programming Systems*, Principal Investigator: Bengt Nordström. Other members of our consortium are actively involved. Technically, this new project will require integrating some of our long standing areas of interest, such as dependent pattern matching, rewriting, termination of recursion, and modular construction. Introducing a new language is certainly risky (little chance of becoming an industrial standard), but there are areas like databases and web-applications where there is a need for special languages (like SQL, PHP, XML) and if security or correctness is an important issue, our dependently typed languages could become important. Progress on the goal will have significant impact on the construction and use of proof tools, as well as on programming in general. Thus we feel that this new work supports our application under the FET Open program.

Another bold idea which our work contributes to is the idea of an Interactive Mathematical Assistant, that incorporates the mathematical activities of defining, proving and computing within a document-editing environment. We will not be able to do this within the Types project, but our work will play an important role in achieving it.

5 Potential impact

Formal and semi-formal reasoning and specification methods are now important in many economic and scientific areas. For example, the Common Criteria for Information Technology Security (ISO International Standard 15408) calls for such methods in its highest Assurance Levels, EAL5–EAL7.[2] Formal methods will undoubtedly take on even greater importance in the future. Europe traditionally has a strong position in the research in this field. The breadth, depth, and long-term committment of our consortium make us very well placed to pursue this area for the benefit of European industry and science. In addition, we have ongoing connections with top US researchers, who are often invited speakers at our workshops and summer schools.

As in all engineering sciences, the impact of our research lies in industry, education and inside science. It is difficult to measure this impact, but our periodic reports will try to give an impression of it. The industrial impact can be seen in how our ideas are used in industry, how industrial people are attending our workshops and schools etc. Educational impact can be seen in how many courses and seminars are given in various universities. Scientific impact can be seen in our influence on the scientific community; how many papers we produce, how many conference presentations we make etc.

We elaborate on our potential impact on **industry** and **science**. The next section will contain a discussion of **dissemination** of our work.

Industry

The difficulty of building large scale computer-based systems that are correctly specified and reliably implement their specifications is a major problem in many economic sectors, from business information systems through air traffic control and fly-by-wire systems. Meeting this challenge requires progress on many fronts, and our proposed project contributes in several ways.

Improved programming languages It is well known that high-level programming languages make a large contribution to software reliability. Their behaviour can be precisely specified, and at the same time is more convenient and intuitive for programmers to use. One of the first applications of our technology of formal reasoning is the analysis of properties of programming languages. This is an established and successful area for our consortium; e.g. members of our consortium have made real contributiuons to the study of Java and JavaCard and functional languages. General features important to many programming languages, such as modularity and subtyping, are also among our topics.

Dependently typed programming For special purposes (high reliability, rapid prototyping, modular program construction), languages with even more expressive type systems, such as dependent types, may prove to be desirable. This is a new and exciting topic, which our consortium is among the best placed European groups to pursue.

Formal specification and proof of program correctness Even with well specified and expressive programming languages, some program systems will require formal specification and proofs of correctness. This is one of the main topics our consortium has pursued. Two approaches

[2] Schlumberger Smart Cards & Terminals recently announced that its development methodology has been evaluated and validated for Common Criteria Evaluation Assurance Level 7. The Schlumberger methodology, which was developed together with Trusted Logic, a member of our consortium, uses a mathematically proven environment for loading, verifying and executing.

that are proving successful in our work are program extraction from a proof that some correctness property can be satisfied, and special purpose tools with built-in knowledge of particular programming languages built on top of general purpose proof tools.

Science

Original mathematics is entirely the business of human mathematicians for the foreseeable future. Nevertheless we see scope for formal reasoning tools to have significant impact on the practice of mathematics, including areas such as presentation, communication, archiving and searching, collaboration, computational experimentation and education.

Computational experimentation Use of computer algebra systems by mathematicians and scientists is now very common. Today, no computer algebra system is foundationally correct; there are many side conditions and invariants that are not checked or maintained. Dependent types are already a significant aspect of most computer algebra systems (sometimes implicitly so), and our dependently typed reasoning tools are well placed to provide semantic foundations for computer algebra. We have already made progress, e.g. proofs of correctness of algorithms. But in the long term we expect to have more impact by integrating foundational proof with correct calculational tools.

Presentation, communication, archiving and searching, collaboration Computer typesetting tools such as LaTeX have had a profound impact on presentation and communication of mathematics and science. Ongoing work by consortium members on representing, not just symbols, but their meaning (MoWGLI, OpenMath, OMDoc, based on XML technologies), will have even greater impact on even more aspects of science. Even without formal proof, "mathematical vernacular" languages supporting formal statement of assumptions, definitions and theorems can underpin internet supported collaboration on building large, distributed and mechanically searchable libraries of mathematical knowledge. This is a new and challenging field, and it is impossible to predict where it will lead.

Education We believe the possibilities of the previous items can also impact mathematics education. Imagine an on-line textbook with conventional, informal presentation. However, students may click on lemmas and definitions to see more precise, formal statements, and may choose to examine proofs in greater detail, going down to completely formal presentation. Similarly, foundational tools (computer algebra, proof checker), integrated with the text, will support experimentation and proof in exercises.

In software engineering it is often pointed out that practicing engineers responsible for large projects won't take the risk of beginning to use new technology based on formal methods. One way to overcome this startup problem is to give students (future practicioners) confidence and proficiency in formal approaches by educational exposure to formal mathematics.

6 Project management and exploitation/dissemination plans

6.1 Project management

The project is headed by the Project Coordinator (PC) Prof.dr. B. Nordström, from Chalmers University, Göteborg Sweden. Decisions concerning the project are taken by the Steering Committee (SC) on the basis of a majority vote. The PC is the "managing director" of the project and will take care of running matters. It is not feasible to have a steering committee consisting of representatives of all main sites, meetings with 15 people have a tendency to become formal and improductive. Therefore we have a rather small committee, it consists of the following members:

- Prof.dr. B. Nordström, Chalmers University, Göteborg Sweden (chairman and PC).
- Prof.dr. Ch. Paulin-Mohring, University Orsay, Paris France.
- Prof.dr. P. Aczel, University of Manchester, UK.
- Prof.dr. Z. Luo, Royal Holloway, University of London, UK.
- Dr. R. Pollack, University of Edinburgh, UK.
- Dr. H. Geuvers, University of Nijmegen, NL.

In all important issues, the SC seeks support from the leaders of the sites and the PC informs the site leaders by e-mail on a regular basis. The SC will have an annual physical meeting, at the annual TYPES workshop. At these workshops there will also be a plenary business meeting to discuss plans with the participants and to solicit ideas from them.

6.2 Plan for using and disseminating knowledge

The work of our consortium is mostly at the research level. It is disseminated by many refereed publications in scientific conferences and journals. The TYPES group has had 10 international workshops, open to interested outsiders, and with invited speakers also from outside the consortium. Each international workshop is followed by a call for papers on appropriate topics, open to outsiders, leading to published proceedings refereed to the highest standard [12, 13, 2, 8, 3, 11, 1, 6, 5, 9]. The current consortium proposes to continue these workshops and refereed proceedings.

To address young researchers, we organize summer schools where we present the research of our consortium. Also various researchers of our consortium act as teachers in other summer schools in Europe and the US to disseminate our work.

Our proof systems (including Coq, Isabelle and Mizar) are a significant means of dissemination. They are all freely available on the internet [16], including documentation, examples, and large and growing libraries of formalised mathematics and computer science. They are widely used by researchers and students, also outside our consortium. Several impressive proof developments have been carried out. A large number of advanced students have used, and contributed to the development of, these systems, and then go on to disseminate this work further in industry and academia.

We also have dissemination activities for industrial needs. Many participating teams have strong collaborations with industrial partners, in the area of critical systems development (smartcard technology for instance) or proof presentation, some of them (France Telecom, Dassault Aviation) being part of the consortium. We shall invite our industrial contacts to participate in annual and thematic workshops, giving them the opportunity to present challenging problems or interesting case studies. In the past, sites have organised training in their tools and methodology in a format suitable for industry (a few days of hands-on tutorial, accessible with no previous theoretical knowledge).

The students we are training are natural candidates for employment in industry specialized in formal methods. More than that, bright students who feel comfortable with a new technology don't just fill the skill needs of industry, they accelerate technology transfer by encouraging their employers to use the technology they are familiar with. Their success in addressing some industrial problems can encourage industrial employers to experiment further with new technology.

Our projects present themselves on web pages [4, 17, 18] that point to other web pages about our activities; e.g. the summer school [15].

6.3 Raising public participation and awareness

As mentioned, our tools are free, publicly available on WWW, and come with substantial libraries, and tutorial and reference manuals. Consequently some of our tools are included in popular software download sites on the web.

Some of our researchers also write popular or non-specialist science articles (e.g. [14, 7]).

We have proposed work towards using our proof tools in wider mathematics education (section 2.2), which addresses an audience outside the community of theoretical computer science and formal methods. Also, our project has a thread of research in natural language technology, along with a freely available tool for working in this field [10].

7 Work plan

7.1 Introduction – general description and milestones

The work plan is structured into five work packages describing the activities of the CA and a list of deliverables allowing for monitoring and reviewing of the progress and success of the CA. The activities of the CA in WP 1 are "management activities". The other WPs, WPs 2–6 are "co-ordination activities": meetings of various type, individual visits, a summer school, short courses and the web site. The site of the project coordinator (PC), site 1, is the lead contractor for all work packages: the PC (together with the SC) coordinates the activities and makes sure that the meetings and summer school are organized and that the web-site gets set up. (See section 6 for details of the management structure.) The individual site leaders have the responsibility to assist the PC and SC in achieving the tasks of the CA, and in particular that the goals of WP 5 (the individual scientific visits) are achieved and that the short courses of WP4 get organized. It is impossible in this early stage to distribute the work of all activities to the site which is going to organize it.

The work plan is centered around the three annual TYPES meetings, which form the main platform for communicating new research ideas.

7.2 Work planning and timetable

The research carried out within the project will be supported from other sources (EU, national or other funds). The financial contribution from the EU for the project will be used to support the following five activities.

1. Three TYPES meetings
2. Thematic workshops
3. Types Summer School
4. Individual visits
5. Types WWW site

Each of these activities have their own management structure, which is detailed below. A preliminary timetable for the major event is shown below:

2004	September	
	October	
	November	
	December	TYPES Meeting 1
2005	January	
	February	Thematic Workshop 1
	March	
	April	
	May	
	June	
	July	Thematic Workshop 2
	August	Summer School
	September	
	October	
	November	
	December	Thematic Workshop 3
2006	January	
	February	
	March	
	April	
	May	TYPES Meeting 2 Thematic Workshop 4
	June	
	July	
	August	
	September	
	October	Thematic Workshop 5
	November	
	December	
2007	January	
	February	
	March	
	April	
	May	TYPES Meeting 3 Thematic Workshop 6
	June	
	July	
	August	
	September	
	October	
	November	
	December	Thematic Workshop 6
2008	January	
	February	Thematic Workshop 7
	March	Thematic Workshop 8
	April	

A timetable for the events and the associated deliverables can be found in Section 7.

1. Three TYPES meetings, WP2
   The first meeting will take place in Paris-Sud in the winter of 2004, and the remaining ones in the spring of 2006 and 2007. The PC will solicit potential organizers and propose them to the SC. The date and place of the annual meeting will be fixed at least half a year before the actual meeting. The PC together with the organizers take responsibility for the scientific program. The meetings are "open", i.e. we explicitly invite persons from outside the project to participate. The PC makes sure that informal proceedings appear on the Types WWW-site soon (3 months) after the meeting and that refereed proceedings of the meetings are published (e.g. [12, 13, 2, 8, 3, 11, 1, 6, 5, 9]).

2. Thematic workshops. WP3
   The dates and the topics of these workshops are not yet completely fixed: for the first year we have planned two workshops (see the Workplan 7), while for the second and third year we have some suggested workshops, but we will solicit ideas from the project members during the project; this may also raise the number of thematic workshops above 6 for the whole project period. The SC will ensure a reasonable distribution of the timing of workshops, and of their topics. The thematic workshops are also open to persons from outside the project. The PC will ensure that proceedings for the thematic workshops will be created in an electronic form and will appear on the Types WWW-site.

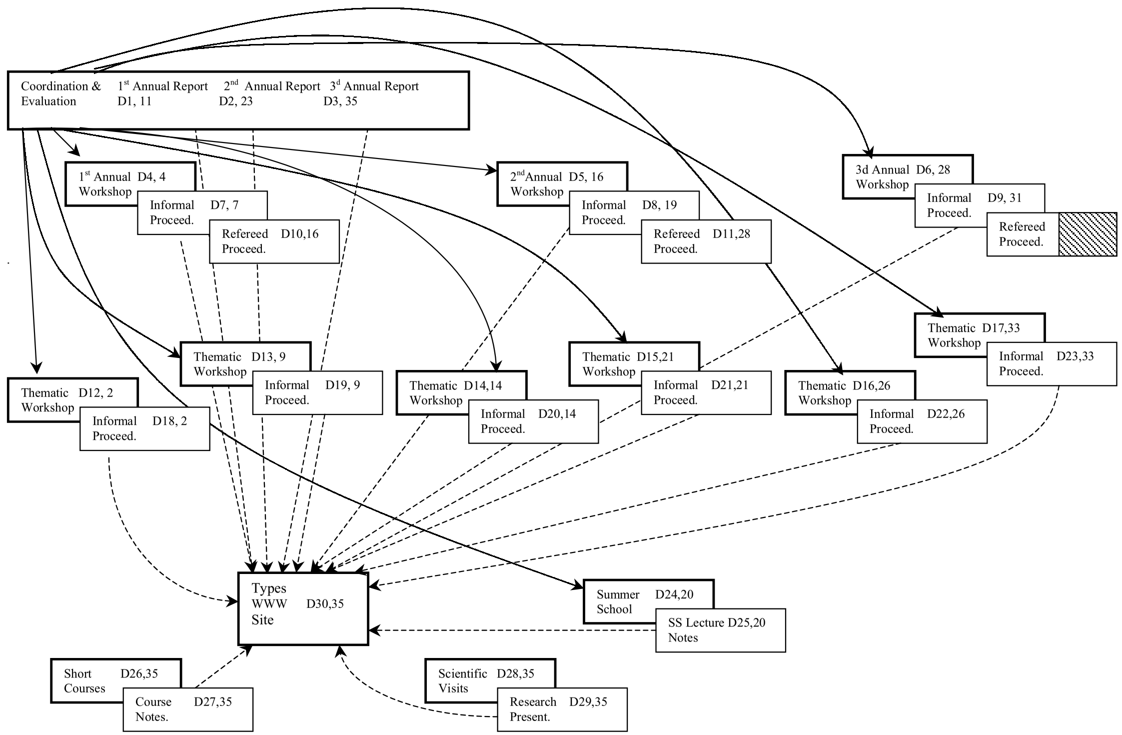
3. Types Summer School. WP4
   In the summer of 2005, a summer school will be organised. The SC will decide upon a place and a date for the summer school, and create an organising committee and a program committee to organise the school. The SC will compose the programme committee in such a way as to encourage depth and breadth of the summer school, seeking to integrate the topics of the project in a wider perspective. The SC will explicitly encourage the participation of researchers from outside the project in this summer school.

4. Individual visits and short courses. WP4 and WP5
   Each site has a budget for short visits. From this budget visits to other sites can be paid and persons from other sites can be invited (and paid) as a visitor. The length of such a visit should be up to 14 days and the hosting site should take care that a research presentation or a short course is given by the visitor during this period. The site leaders are themselves responsible for these visits. Each visit will be communicated to the PC. Course notes of the short courses will be communicated to the PC and be made public on the Types WWW-site.

5. Types WWW-site. WP6
   A Types website will be created that gives an up-to-date account of the state of the project: its members, coming and past activities (workshops, summer school), visits and related events. Furthermore, all informal proceedings, course notes and lecture notes will be made available through the WWW-site. Also a types mailing list will be created. The PC is responsible for the creation and maintenance of the WWW-site.

7.3 Graphical presentation of the components

On the following page we have presented a pert diagram to show the interdependencies of the various activities and deliverables.



A normal arrow indicates an activity that is actively initiated by the project coordinator (PC) and the steering committee (SC). A dashed arrows indicate documents that will appear on the WWW-site as (part of) deliverables. Of course, also announcements and programs of the annual meetings, the thematic workshops, the summer school etc. will appear on the WWW-site, but this is not explicitly indicated in the diagram.

7.4 Work package list

7.4. Work package list

Work package No	Work package title	Lead contractor No	Person months	Start month	End month	Deliverable No
WP1	Coordination and Evaluation	1	N/A	0	43	1–3
WP2	TYPES meeting	1	N/A	0	43	4–11
WP3	Thematic workshops	1	N/A	0	43	12–23 , 32 – 37
WP4	Education	1	N/A	0	43	24–27
WP5	Individual visits	1	N/A	0	43	28,29
WP6	The TYPES web page and mailing list	1	N/A	0	43	30

NB The duration of all work packages is 44 months, so they run from the beginning until the end of the project (month 0 – month 43). Specific events (like workshops) within these work packages have a specific delivery date, as can be seen from the deliverables list, but in the "duration" of the work packages we take into account the time spent on the organization and preparation of events. This will start as soon as the project is granted.

7.5 Deliverables list

7.5. Deliverables list

We have fixed the organization of the events in specific months. Month 1 is the first month of the project. It is possible that this has to be revised depending on the start date of the project, we want for instance the summer school to take place in the summer. If, in the course of the project, a specified month for an event turns out to be impossible or inconvenient, we allow ourselves to move the event by at most three months.

The leading participant of most deliverables has not yet been decided. It is the responsibility of the coordinating site (together with the steering committee) to decide this during the course of the project. It is of this reason that we have put the coordinator as the leading partner in most events. It is not practical to decide at this early stage the topic (and therefore the leading partner) of the thematic workshops.

Del. no.	Deliverable name	WP no.	Lead part.	Estim. mnths.	Nature	Diss. level	Deliv. date
D1	Periodic Project Report 1	1	1		R	PU	11
D2	Periodic Project Report 2	1	1		R	PU	23
D3	Periodic Project Report 3	1	1		R	PU	35
D4	TYPES meeting 1	2	2		O	PU	4
D7	Informal Proc. of TYPES meeting 1	2	2		R	PU	7
D10	Refereed Proc. of TYPES meeting 1	2	2		R	PU	16
D5	TYPES meeting 2	2	1		O	PU	21
D8	Informal Proc. of TYPES meeting 2	2	1		R	PU	24
D11	Refereed Proc. of TYPES meeting 2	2	1		R	PU	33
D6	TYPES meeting 3	2	1		O	PU	33
D9	Informal Proc. of TYPES meeting 3	2	1		R	PU	36
D12	Thematic workshop 1	3	1		O	PU	6
D18	Proceedings of workshop 1	3	1		R	PU	6
D13	Thematic workshop 2	3	1		O	PU	11
D19	Proceedings of workshop 2	3	1		R	PU	11
D14	Thematic workshop 3	3	1		O	PU	16
D20	Proceedings of workshop 3	3	1		R	PU	16
D15	Thematic workshop 4	3	1		O	PU	21
D21	Proceedings of workshop 4	2	1		R	PU	21
D16	Thematic workshop 5	3	1		O	PU	26
D22	Proceedings of workshop 5	3	1		R	PU	26
D17	Thematic workshop6	3	1		O	PU	33
D23	Proceedings of workshop 6	3	1		R	PU	33
D24	Summer school	4	1		O	PU	12
D25	Lecture notes for summer school	4	1		R	PU	12
D26	Short courses	4	1		O	PU	43
D27	Course notes for short courses.	4	1		R	PU	43
D28	Individual scientific visits	5	1		O	PU	43
D29	Research presentations during sci. visits	5	1		O	PU	43
D30	WWW-site	6	1		O	PU	43
D31	CD	6	1		O	PU	43
D32	Thematic workshop 7	3	1		O	PU	39
D33	Proceedings of workshop 7	3	1		R	PU	39
D34	Thematic workshop 8	3	1		O	PU	41
D35	Proceedings of workshop 8	3	1		R	PU	41
D36	Thematic workshop 9	3	1		O	PU	43
D37	Proceedings of workshop 9	3	1		R	PU	43

7.6 Work package description

7.6. Work package description

Work package description: Coordination and evaluation WP1

Work package number:	WP1														
Starting date:	Month 0														
Activity Type:	Management activities														
Participant id:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Person-months per participant:	6	.5	.5	2	.5	.5	2	.5	2	2	2	.5	.5	.5	.5

Objectives

- Coordination and management of the CA
- Administration, reporting to and contacts with the EU.

Description of work
This work package consists of the management of the CA. Apart from the overall responsibility for the work packages and the deliverables, the work comprises

- Contact person for the EU
- Responsibility for the periodic progress reports
- Coordination of the management structure (see Section 6).

Deliverables

- Periodic Project Reports (D1,D2,D3)

Milestones and expected results Periodic Project Reports in months 11, 23 and 35.

Work package description: TYPES meeting WP2

Work package number:	WP2														
Starting event:	Types Meeting in Dec 2004														
Activity Type:	Coordination activities														
Participant id:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Person-months per participant:	18	12	27	15	9	12	9	9	24	9	12	12	12	12	3

Objectives

- Presentation and exchange of results obtained in the working group.
- Opportunity for communication between researchers from different research groups and between researchers working in different fields, especially between theorists and practitioners.
- Catalyzing collaboration.

Description of work
This work package comprises three TYPES meetings. The first of these workshops will be held in December 2004. The second will be held in spring 2006. The third will be held in spring 2007. Each meeting will be open to all participants of the project (including representatives from industrial partners) and also to interested researchers from outside the project.

Each meeting will last for four days and consist of 2-3 invited talks, 20-30 contributed talks within the various themes and 5-10 system demonstrations. The program committee will make a selection of the proposed talks and system demonstrations. In addition there will be a panel discussion, a business meeting (open to members), and a meeting of the steering committee.

Three months after each meeting, informal proceedings consisting of contributed papers and transparencies will be made available on our WWW site. Also, after the workshop, a refereed proceedings of selected papers will be published as a Springer LNCS or in a journal. The publication of the refereed proceedings for the third workshop will fall outside of the period of the CA and therefore it is not listed as a deliverable. An electronic version will be available on our web page.

Deliverables

- TYPES meeting (D4, D5, D6)
- Informal TYPES proceedings (D7, D8, D9)
- Refereed TYPES proceedings (D10, D11)

Milestones and expected results TYPES meetings in months 4, 21 and 33.
Proceedings will be available earlier than three months after the meeting through the WWW-site.
Refereed proceedings of selected papers approx. 12 months after the meeting.

Work package description: Thematic workshops WP3

Work package number:	WP3														
Starting event:	Workshop “Libraries of Formalized Mathematics”, fall 2004														
Activity Type:	Coordination activities														
Participant id:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Person-months per participant:	12	8	18	10	6	8	6	6	16	6	8	8	8	8	2

Objectives

- Bringing together researchers working on a specific topic.
- Promoting the development and exploration of new research fields.

Description of work During the project we will hold small specialized workshops, typically two per year, on selected thematic topics. We have planned two workshop topics in the first year. However, we don’t want to fix the topics of the workshops in years 2 and 3 at this time, because new topics may arise among the participants that are considered to be of higher priority. The SC will decide on the actual topics of the workshops, soliciting input from the site leaders (by e-mail) and the participants (at the annual business meetings). We encourage the participants to organize more than 6 workshops, if the need is felt for that.

Workshops in year 1.

- Libraries of Formalized Mathematics
- Programming with Dependent Types

Possible topics for Workshops in year 2-3.

- High-level languages for Proof description (Mizar, Isar, Ltac, ...)
- Interfaces of Proof Assistants
- Constructive Analysis and Program Extraction
- Formal Topology (in type theory) and Constructive Topology (in set theory)
- Intensional Lambda Calculi and Modal Type Theory
- Math Wiki
- Dependently Typed Programming
- Program Extraction from Proofs

Deliverables

- Thematic workshops (D12, D13, D14, D15, D16, D17 ,D32, D34, D36)
- Thematic workshop proceedings, available at the workshops and through our WWW-site (D18, D19, D20, D21, D22, D23 , D33, D35, D37)

Milestones and expected results A minimum of six workshops in months 6, 11, 16, 21, 26 and 33.
Additional workshops will take place in months 39, 41 and 42
Proceedings of all workshops available through the WWW-site.

Work package description: Education WP4

Work package number:	WP4														
Starting date:	Month 0														
Activity Type:	Training activities														
Participant id:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Person-months per participant:	6	4	9	5	3	4	3	3	8	3	4	4	4	4	1

Objectives

- Disseminating the research of the working group
- Training of young researchers

Description of work We will organize a summer school open to postgraduate students and researchers in the summer 2005. The goals of this school are to disseminate the research results obtained in the working group and to educate young researchers broadly in our field. The lecture notes of the summer school will be made available through our website and as a technical report.

In addition to the summer school we envisage short courses to be held by visiting researchers at their host site on their individual research topics. These course notes will be made available through our WWW-site.

Finally, two sites plan to publish overview books in our field: the INRIA Sophia-Antipolis site on the use of the system Coq and its type theory and the Nijmegen site on type theory in general. These are not deliverables of the present project, but the project and its meetings will support the interaction and cooperation necessary for these achievements.

Deliverables

- Organization of summer school (D24)
- Lecture notes of the Summer School (D25)
- Organization of short courses (D26)
- Course notes of short courses (D27)

Milestones and expected results

- Summer school in month 12. Lecture notes for this school available at the same time, also through our WWW-site.
- Short courses throughout the project. Course notes are made available, also through our WWW-site.

Work package description: Individual visits WP5

Work package number:	WP5														
Starting date:	Month 0														
Activity Type:	Coordination activities														
Participant id:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Person-months per participant:	12	8	18	10	6	8	6	6	16	6	8	8	8	8	2

Objectives

- Enabling collaboration between individual researchers on a particular topic.
- Cross fertilization of research.
- Creating the possibility for short courses by visiting researchers.

Description of work
We will partially fund individual short-term visits between sites to provide opportunities for collaborative research and training of junior researchers and to enable cooperation between senior members. The results of any such visit will be summarized in the progress report. Such visits will only be funded if the visiting researcher presents a short course or a research talk at the hosting institution, see D26.

Deliverables

- Individual scientific visits (D28)
- Research Presentations (D29)
- Short courses. (D26)
- Course notes of short courses. (D27)

Milestones and expected results We expect a minimum of 10 visits per year and one research presentation or short course during such a visit. These will be reported in the progress reports to be delivered at months 12, 24 and 36. The course notes will be available through our WWW-site.

Work package description: The TYPES web page, CD and mailing list, WP6

Work package number:	WP6														
Starting date:	Month 0														
Activity Type:	Coordination activities														
Participant id:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Person-months per participant:	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Objectives

- Dissemination and information

Description of work

We will maintain a project web page, containing (links to) all (other) deliverables and results as well as other project-related documentation (announcements of events etc.) and serving as a "types portal" for the wider community. The page will contain links to all sites and sub-sites and all sites and sub-sites will provide a link to the project web page.

The web-page will be frozen on a CD at the end of the TYPES project. This will contain all project deliverables, publications and important results.

We will also maintain a "types working group" mailing list of all actual participants of the project. This will be used to communicate with the participants in a fast and efficient way (announcements, calls etc.).

Deliverables

- WWW-site (D30)
- CD (D31)

Milestones and expected results The work on the WWW-site will be started in month 0. At its latest from month 1 onward we will have the TYPES web page and the mailing list operational. It will be continuously updated with relevant information.

7.6.1. Description of the Deliverables

The periodic progress reports (deliverables D1, D2 and D3) are produced by the site leaders and the Project Coordinator.

Each of the annual general workshops (deliverables D4, D5 and D6) will be hosted by a site to be selected by consultation between the Steering Committee and site leaders.

The refereed proceedings of the annual general workshops (deliverables D10 and D11[3]) are edited by leading researchers from the site that hosted the meeting. This includes finding a publisher, soliciting fully developed papers, organising independent refereeing, selecting papers on the basis of referee reports, and seeing the publication to completion.

The thematic workshops (deliverables D12, D13, D14, D15, D16 and D17) are hosted by sites which volunteer, because they want to encourage and collaborate on research into a particular theme.

The informal proceedings of all workshops, thematic and general (deliverables D7, D8, D9, D18, D19, D20, D21, D22 and D23), are edited and produced (on WWW and/or as technical reports) by researchers from the hosting site.

The summer school (deliverable D24) will be hosted by a site to be selected by consultation between the Steering Committee and site leaders. The lecture notes of the summer school (deliverable D25) are produced by the lecturers and invited speakers. They are collected on a web site by the hosting site.

D1. Periodic progress report in month 11. This will contain a description of the activities in the project and reports on achievements/benefits of all these, including individual visits, courses and workshops.

D2. Periodic progress report in month 23. This will contain a description of the activities in the project and reports on achievements/benefits of all these, including individual visits, courses and workshops.

D3. Periodic progress report in month 35. This will contain a description of the activities in the project and reports on achievements/benefits of all these, including individual visits, courses and workshops.

D4. Types meeting in month 4

D5. Types meeting in month 21.

D6. Types meeting in month 33.

D7. Informal proceedings of the Types meeting in month 7, D4 to be published on the project WWW-site. (Later, a refereed proceedings of selected papers is to be published as an LNCS or a special issue of an appropriate scientific journal, deliverable D10).

D8. Informal proceedings of the Types meeting in month 24, D5 to be published on the project WWW-site. (Later, a refereed proceedings of selected papers is to be published as an LNCS or a special issue of an appropriate scientific journal, deliverable D11).

[3]Refereed proceedings of the final annual general workshop will appear after the Coordination Action is completed, so is not listed as a deliverable.

D9. Informal proceedings of the Types meeting in month 36, D6 to be published on the project WWW-site. (Later, a refereed proceedings of selected papers is to be published as an LNCS or a special issue of an appropriate scientific journal. These will appear after the closing date of the CA and hence these proceedings are *not* presented as a deliverable.)

D10. Refereed proceedings of selected papers from the first Types meeting, D4, to be published as an LNCS or a special issue of an appropriate scientific journal.

D11. Refereed proceedings of selected papers from the second Types meeting, D5, to be published as an LNCS or a special issue of an appropriate scientific journal.

D12. Thematic workshop 1. "Libraries of Formalized Mathematics"

D13. Thematic workshop 2. "Programming with Dependent Types"

D14. Thematic workshop 3. A minimum of six thematic workshops will be held with a minimum of two per year. The topics of the first two workshops are fixed; for the topics of workshop 3–6 we have a list of suggestions, but these are open for initiatives from the project participants. For more details see the work package WP3.

D15. Thematic workshop 4.

D16. Thematic workshop 5.

D17. Thematic workshop 6.

D18. Proceedings of thematic workshop 1, D12. For each thematic workshop proceedings will be made available on our WWW-site and where appropriate polished proceedings will appear in scientific fora.

D19. Proceedings of thematic workshop 2, D13.

D20. Proceedings of thematic workshop 3, D14.

D21. Proceedings of thematic workshop 4, D15.

D22. Proceedings of thematic workshop 5, D16.

D23. Proceedings of thematic workshop 6, D17.

D24. Summer school.

D25. Lecture notes of the summer school, D24, made available at the WWW-site.

D26. Short courses held by various (visiting) researchers.

D27. Published course notes of courses.

D28. Scientific visits. These will be announced through the WWW-site.

D29. Research presentations during scientific visits. Visiting scientists should give a presentation during their visit. These will be announced through the WWW-site.

D30. Project WWW-site. Throughout the project we will maintain a WWW site containing (links to) all other deliverables as well as other project-related information such as links to partners, proposal text, timetable, planned activities. This WWW site will also serve as a "types portal" giving access to relevant on-line resources such as homepages of researchers, events calendar, bibliography, tutorials and surveys, etc. We employ a part-time worker whose main job is to maintain and develop the WWW site.

D31. Project CD. In the end of the project, we will freeze the WWW-site on a CD/DVD. This will then contain all project deliverables, publications and important results.

D32. Thematic workshop 7.

D33. Proceedings of thematic workshop 7, D32.

D34. Thematic workshop 8.

D35. Proceedings of thematic workshop 8, D32.

D36. Thematic workshop 9.

D37. Proceedings of thematic workshop 9, D32.

8 Project resources and budget overview

8.1 Efforts for the full duration of the project

The following page shows how many person-months each participant is using in each activity.

CA Project Effort Form
Full Duration of project

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
COORDINATION activities																
WP2 Annual Meetings	18.0	12.0	27.0	15.0	9.0	12.0	9.0	9.0	24.0	9.0	12.0	12.0	12.0	12.0	3.0	195.0
WP3 Thematic Workshops	12.0	8.0	18.0	10.0	6.0	8.0	6.0	6.0	16.0	6.0	8.0	8.0	8.0	8.0	2.0	130.0
WP4 Education	6.0	4.0	9.0	5.0	3.0	4.0	3.0	3.0	8.0	3.0	4.0	4.0	4.0	4.0	1.0	65.0
WP5 Individual Visits	12.0	8.0	18.0	10.0	6.0	8.0	6.0	6.0	16.0	6.0	8.0	8.0	8.0	8.0	2.0	130.0
WP6 WWW site	4.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4.0
TOTAL Coordination activities	52.0	32.0	72.0	40.0	24.0	32.0	24.0	24.0	64.0	24.0	32.0	32.0	32.0	32.0	8.0	524
MANAGEMENT activities																
WP1 Coordination& Evaluation	6.0	0.5	0.5	2.0	0.5	0.5	2.0	0.5	2.0	2.0	2.0	0.5	0.5	0.5	0.5	20.5
TOTAL activities	58.0	32.5	72.5	42.0	24.5	32.5	26.0	24.5	66.0	26.0	34.0	32.5	32.5	32.5	8.5	544.5

1. Chalmers Göteborg
2. CNRS Paris 7
3. INRIA
4. Paris Sud
5. LMU Munich
6. TU Munich
7. Nijmegen
8. Białystok
9. Durham - Royal Holloway
10. Edinburgh
11. Manchester
12. Torino
13. Udine
14. Warsaw
15. Tallin

The table displays the amount of personmonths that the various sites (incl. their subsites) will spend on organising of and participating in events.

8.2 Overall budget for the full duration of the project

The project will run for three years and will cost **EUR 647 000**. We estimate that the direct costs will be **EUR 539 170**. The indirect costs are 20 % of this. The direct costs have been obtained by adding the following components:

Total direct cost for 14 main sites: 254 800
Each main site will each year get money for three persons attending the TYPES meetings and one person attending a thematic workshop and one individual visit. This will make 3 years * (3*1250 + 1 000 +1 320) = 18 200 for each site. For 14 sites this will be 254 800.

Direct cost for IOC Tallinn 15 267
Tallinn is a small main site and will get 83 % of a main site = 15 267.

Direct cost for INRIA Futurs subsite: 18 200
INRIA Futurs is costed as a main site, and is proposed as a subsite only for INRIA internal management reasons.

Direct cost for 19 subsites (excluding INRIA Futurs): 172 900
Each subsite will get half of a main site = 9 100. For 19 subsites this will be 172 900.

Additional cost for the TYPES meetings: 18 000
Each meeting will have three invited speakers, 1 meeting * 3 years * 3 speakers * 2 000 = 18 000.

Additional cost for the thematic workshops: 11 000
Each meeting will have one invited speaker, 2 meetings * 3 years * 1 speaker * 1 833 = 11 000.

Total cost for the summer school: 34 000
There will be 8 invited lecturers and 30 student grants, this makes 8*2000 + 30 * 600 = 34 000.

Coordination costs at Chalmers: 15 000
This consists of salary for a senior researcher for coordination activities, salary for secretarial help (with correspondence and the Types homepage), desktop computer and consumables.

Total direct cost: 539 167

Total direct and indirect cost: 647 000.

Cost breakdowns The following table shows how we have calculated the costs for one person in different activities:

	hotel	travel	fee	expenses	total
Types Meeting					
- participant	600	500	150	-	1250
- invited	600	1000	150	250	2000
Thematic workshop					
- participant	400	500	100	-	1000
- invited	433	1000	150	250	1833
Summer school					
- lecturer	600	1000	150	250	2000
Individual visit	600	500	-	220	1320

8.3 Management level description of resources and budget

The main aspects covered by this Coordination Action are workshops, a summer school, and individual travel between sites for collaborative work. These events support research collaboration and training in our consortium, which would be much reduced without this support. Other deliverables, such as refereed proceedings and lecture notes, follow from these events.

Thus the budgeted direct costs are almost entirely devoted to travel and accomodation at the various workshops and sites. This is computed by allocating each main site a certain number of people participating in the TYPES annual meeting, in thematic workshops and individual visits, per year of the project. (Subsites, which are smaller research groups, are allocated half as many participants.) We have estimated costs for each type of participation. Each of the work packages are included in this analysis (although the costs are budgeted by site participation, rather than by work package), and are supported to a level that has produced successful collaboration and deliverables in past EU-funded TYPES projects.

There are additional costs budgeted for some work packages, covering travel and accomodation of invited speakers at annual meetings, thematic workshops and the summer school. There are also some costs budgeted for the coordinating site, Chalmers, covering coordinating activities, including the TYPES web page deliverable.

The research group in Durham moved to Royal Holloway two months after the project started. The budget of Durham also moved.

8.4 Budget breakdown

Distribution of money over the sites The coordinating site (Chalmers) will adminstrate the additional costs for the following activities:

1. Summer school: invited lecturers and student grants
2. Workshops: invited speakers
3. Types Meeting: invited speakers

Every site will adminstrate the site specific cost for its subsites. This is shown in figure 1.

Distribution of money over time The main expenses for this consortium are in connection with the Types meetings (in the 1:st, 4:th and 6:th period) and the Summer school (in the 2:nd period). We show in figure 2 how the distribution is suggested. Most of the site specific costs (80 %) are in

General costs:		direct costs	indirect	direct + indirect
School		34000	6800	40800
Workshops		11000	2200	13200
Types meeting		18000	3600	21600
total		63000	12600	75600
Coordination		15000	3000	18000
total general cost		78000	15600	93600
Site specific costs:	nr of subsites	site spec	indirect	
1 Chalmers	3	45500	9100	54600
2 Paris 7	1	27300	5460	32760
3 INRIA	5	63700	12740	76440
4 UPS	2	36400	7280	43680
5 LMU	0	18200	3640	21840
6 TUM	1	27300	5460	32760
7 KUN	0	18200	3640	21840
8 Bialystok	0	18200	3640	21840
9 Durham	0	0	0	0
10 Edinburgh	0	18200	3640	21840
11 Manchester	1	27300	5460	32760
12 Torino	1	27300	5460	32760
13 Udine	1	27300	5460	32760
14 Warsaw	1	27300	5460	32760
15 Tallinn	0	15267	3053	18320
16 Royal Holloway	5	63700	12740	76440
total:		461167	92233	553400
General + specific cost:		539167	107833	647000
		gen+-spec	ind cost	
Chalmers total:		123500	24700	148200

Figure 1: Budget: Distributed over all sites

connection with the Types meetings. The last column and last row of this figure are also presented on the following pages as a copy of the Contract Preparation Form A3.1 and A3.2.

Period: Activity:		9.04-2.05 Types meeting	3.05-8.05 Summer school	Year 1	9.05-2.06	3.06-8.06 Types meeting	Year 2	9.06-2.07	3.07-4.08 Types meeting	Year 3	Total
1 Chalmers		28460	51140	79600	8090	26210	34300	8090	26210	34300	148200
site spec		14560	3640	18200	3640	14560	18200	3640	14560	18200	54600
Coordination		4500	4500	9000	2250	2250	4500	2250	2250	4500	18000
managment		9400	43000	52400	2200	9400	11600	2200	9400	11600	75600
	school		40800	40800							40800
	workshops	2200	2200	4400	2200	2200	4400	2200	2200	4400	13200
	types meeting	7200		7200		7200	7200		7200	7200	21600
2 Paris 7		8736	2184	10920	2184	8736	10920	2184	8736	10920	32760
3 INRIA		20384	5096	25480	5096	20384	25480	5096	20384	25480	76440
4 UPS		11648	2912	14560	2912	11648	14560	2912	11648	14560	43680
5 LMU		5824	1456	7280	1456	5824	7280	1456	5824	7280	21840
6 TUM		8736	2184	10920	2184	8736	10920	2184	8736	10920	32760
7 KUN		5824	1456	7280	1456	5824	7280	1456	5824	7280	21840
8 Bialystok		5824	1456	7280	1456	5824	7280	1456	5824	7280	21840
9 Durham		0	0	0	0	0	0	0	0	0	0
10 Edinburgh		5824	1456	7280	1456	5824	7280	1456	5824	7280	21840
11 Manchester		8736	2184	10920	2184	8736	10920	2184	8736	10920	32760
12 Torino		8736	2184	10920	2184	8736	10920	2184	8736	10920	32760
13 Udine		8736	2184	10920	2184	8736	10920	2184	8736	10920	32760
14 Warsaw		8736	2184	10920	2184	8736	10920	2184	8736	10920	32760
15 Tallinn		4885	1221	6106	1221	4885	6106	1221	4885	6106	18320
16 Royal Holloway		20384	5096	25480	5096	20384	25480	5096	20384	25480	76440
Total		161473	84393	245866	41343	159223	200566	41343	159223	200566	647000

Figure 2: Budget: Distribution over three years

Contract Preparation Forms

EUROPEAN COMMISSION
6th Framework Programme on Research, Technological Development and Demonstration

Coordination Action

A3.1

Please use as many copies of form A3.1 as necessary for the number of partners

Proposal Number	510996	Proposal Acronym	TYPES

Financial information - whole duration of the project

Participant n?	Organisation short name	Cost model used	Estimated eligible costs and requested EC contribution (whole duration of the project)		Costs and EC contribution per type of activities: Coordination activities (1)	Training activities (2)	Consortium Management activities (3)	Total (4)=(1)+(2)+(3)	Total receipts
1	CHALMERS	AC	Eligible costs	*Direct Costs (a)*	108.500,00	,00	15.000,00	123.500,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	21.700,00	,00	3.000,00	24.700,00	
				Total eligible costs (a)+(b)	130.200,00	,00	18.000,00	148.200,00	
			Requested EC contribution		130.200,00	,00	18.000,00	148.200,00	
2	PARIS 7	AC	Eligible costs	*Direct Costs (a)*	27.300,00	,00	,00	27.300,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	5.460,00	,00	,00	5.460,00	
				Total eligible costs (a)+(b)	32.760,00	,00	,00	32.760,00	
			Requested EC contribution		32.760,00	,00	,00	32.760,00	
3	INRIA	FC	Eligible costs	*Direct Costs (a)*	63.700,00	,00	,00	63.700,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	12.740,00	,00	,00	12.740,00	
				Total eligible costs (a)+(b)	76.440,00	,00	,00	76.440,00	
			Requested EC contribution		76.440,00	,00	,00	76.440,00	
4	UPS	FCF	Eligible costs	*Direct Costs (a)*	36.400,00	,00	,00	36.400,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	7.280,00	,00	,00	7.280,00	
				Total eligible costs (a)+(b)	43.680,00	,00	,00	43.680,00	
			Requested EC contribution		43.680,00	,00	,00	43.680,00	
5	LMU-MUENCH	AC	Eligible costs	*Direct Costs (a)*	18.200,00	,00	,00	18.200,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	3.640,00	,00	,00	3.640,00	
				Total eligible costs (a)+(b)	21.840,00	,00	,00	21.840,00	
			Requested EC contribution		21.840,00	,00	,00	21.840,00	
6	TUM	AC	Eligible costs	*Direct Costs (a)*	27.300,00	,00	,00	27.300,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	5.460,00	,00	,00	5.460,00	
				Total eligible costs (a)+(b)	32.760,00	,00	,00	32.760,00	
			Requested EC contribution		32.760,00	,00	,00	32.760,00	

Contract Preparation Forms

EUROPEAN COMMISSION
6th Framework Programme on Research, Technological Development and Demonstration

Coordination Action

A3.1

Please use as many copies of form A3.1 as necessary for the number of partners

Proposal Number	510996	Proposal Acronym	TYPES

Financial information - whole duration of the project

Participant n?	Organisation short name	Cost model used	Estimated eligible costs and requested EC contribution (whole duration of the project)		Costs and EC contribution per type of activities: Coordination activities (1)	Training activities (2)	Consortium Management activities (3)	Total (4)=(1)+(2)+(3)	Total receipts
7	KUN	AC	Eligible costs	*Direct Costs (a)*	18.200,00	,00	,00	18.200,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	3.640,00	,00	,00	3.640,00	
				Total eligible costs (a)+(b)	21.840,00	,00	,00	21.840,00	
			Requested EC contribution		21.840,00	,00	,00	21.840,00	
8	UWB	AC	Eligible costs	*Direct Costs (a)*	18.200,00	,00	,00	18.200,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	3.640,00	,00	,00	3.640,00	
				Total eligible costs (a)+(b)	21.840,00	,00	,00	21.840,00	
			Requested EC contribution		21.840,00	,00	,00	21.840,00	
9	RISE	AC	Eligible costs	*Direct Costs (a)*	,00	,00	,00	,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	,00	,00	,00	,00	
				Total eligible costs (a)+(b)	,00	,00	,00	,00	
			Requested EC contribution		,00	,00	,00	,00	
10	UEDIN	AC	Eligible costs	*Direct Costs (a)*	18.200,00	,00	,00	18.200,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	3.640,00	,00	,00	3.640,00	
				Total eligible costs (a)+(b)	21.840,00	,00	,00	21.840,00	
			Requested EC contribution		21.840,00	,00	,00	21.840,00	
11	UOM	AC	Eligible costs	*Direct Costs (a)*	27.300,00	,00	,00	27.300,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	5.460,00	,00	,00	5.460,00	
				Total eligible costs (a)+(b)	32.760,00	,00	,00	32.760,00	
			Requested EC contribution		32.760,00	,00	,00	32.760,00	

Contract Preparation Forms

EUROPEAN COMMISSION

6th Framework Programme on Research, Technological Development and Demonstration

Coordination Action

A3.1

Please use as many copies of form A3.1 as necessary for the number of partners

Proposal Number	510996	Proposal Acronym	TYPES

Financial information - whole duration of the project

Participant n?	Organisation short name	Cost model used	Estimated eligible costs and requested EC contribution (whole duration of the project)		Costs and EC contribution per type of activities: Coordination activities (1)	Training activities (2)	Consortium Management activities (3)	Total (4)=(1)+(2)+(3)	Total receipts
12	UNIVERSITA DEGLI STUDI DI TORINO	AC	Eligible costs	*Direct Costs (a)*	27.300,00	,00	,00	27.300,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	5.460,00	,00	,00	5.460,00	
				Total eligible costs (a)+(b)	32.760,00	,00	,00	32.760,00	
			Requested EC contribution		32.760,00	,00	,00	32.760,00	
13	UNIUD	AC	Eligible costs	*Direct Costs (a)*	27.300,00	,00	,00	27.300,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	5.460,00	,00	,00	5.460,00	
				Total eligible costs (a)+(b)	32.760,00	,00	,00	32.760,00	
			Requested EC contribution		32.760,00	,00	,00	32.760,00	
14	UN-W	AC	Eligible costs	*Direct Costs (a)*	27.300,00	,00	,00	27.300,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	5.460,00	,00	,00	5.460,00	
				Total eligible costs (a)+(b)	32.760,00	,00	,00	32.760,00	
			Requested EC contribution		32.760,00	,00	,00	32.760,00	
15	IoC Tallinn	AC	Eligible costs	*Direct Costs (a)*	15.267,00	,00	,00	15.267,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	3.053,00	,00	,00	3.053,00	
				Total eligible costs (a)+(b)	18.320,00	,00	,00	18.320,00	
			Requested EC contribution		18.320,00	,00	,00	18.320,00	
16	RHUL	AC	Eligible costs	*Direct Costs (a)*	63.700,00	,00	,00	63.700,00	,00
				of which subcontracting	,00	,00	,00	,00	
				Indirect costs (b)	12.740,00	,00	,00	12.740,00	
				Total eligible costs (a)+(b)	76.440,00	,00	,00	76.440,00	
			Requested EC contribution		76.440,00	,00	,00	76.440,00	
TOTAL			Eligible costs		**629.000,00**	**,00**	**18.000,00**	**647.000,00**	**,00**
			Requested EC contribution		**629.000,00**	**,00**	**18.000,00**	**647.000,00**	

Contract Preparation Forms

EUROPEAN COMMISSION

6th Framework Programme on Research, Technological Development and Demonstration

Coordination Action

A3.2

Proposal Number	510996	Proposal Acronym	TYPES

Estimated breakdown of the EC contribution per reporting period

Reporting Periods	Start month	End month	Estimated Grant to the Budget: Total	In which first six months
Reporting Period 1	1	12	245.867,00	,00
Reporting Period 2	13	24	200.567,00	41.343,00
Reporting Period 3	25	44	200.566,00	41.343,00
Reporting Period 4			,00	,00
Reporting Period 5			,00	,00
Reporting Period 6			,00	,00
Reporting Period 7			,00	,00

A Consortium description

The consortium brings together groups which are internationally recognised for major contributions to fundamental aspects of Type Theory, for experimentation with innovative concepts in prototypes, and for the development of widely used proof-assistants. Industrial Research and Development teams participate as subsites. They contribute to the application of our tools and methods in industrial case studies.

The past TYPES projects have generated many exchanges of doctoral and post-doctoral researchers between sites. Further, as many of our past PhD students become established researchers, our domain of research has expanded to new universities in new countries. This explains the large size of our consortium.

Both the development of widely-used proof assistants and their application to specialised domain areas require a long-term effort, large human-resources and different skills: theory, proof technology and domain application. The exchanges between sites of people specialised in these different aspects are essential to obtain visible achievements. An example is the complete proof of the Fundamental Theorem of Algebra, it was completed in the previous TYPES project but is still a major source of interaction between sites concerning proof presentation, extraction, real numbers representation.

A.1 Participants and consortium

The consortium consists of 36 research groups from universities and industries in Europe. It would have been unfeasible to let each of these groups be full participants in the action. In order to manage this, we have created a two-level hierachy with 15 main sites and 21 subsites. The money of the consortium is first divided among the main sites, they are then further distributed to the subsites belonging to each main site. It is the responsibility of each main site to make sure that their subsites particpate in and share the resources of the consortium. The subsites are considered as third parties contributing with their expertise to the project. There is a prior agreement between the contractors and the third parties, reflecting their contribution to the project as described below in the presentation of the subsites. We have sent a copy of email from each of the subsites to the Commission expressing this.

The consortium consists of the sites and subsites shown in figure 3. In the following pages we outline the expertise of each (sub)site, and its proposed contribution to the objectives of this action. When it comes to performing the actions of this proposal, there is no distinction between the different partners. Both sites and subsites have equal responsibilities when it comes to the organisation and participation in workshops, summer school and individual visits.

Main site	Subsite	Contact Person
Chalmers University, Göteborg		Nordström
	Bergen University	Bezem
	University of Helsinki	von Plato
	University of Stockholm	Martin-Löf
CNRS/Université Paris 7		Parigot
	Université de Savoie	Raffalli
INRIA Sophia Antipolis		Despeyroux
	INRIA Futurs	Dowek
	University of Bologna	Asperti
	Dassault-Aviation	Ledinot
	University of Minho	Pinto
Université Paris Sud		Paulin-Mohring
	University of Grenoble	Monin
	France Telecom	Alvarado
LMU München		Schwichtenberg
TU München		Nipkow
	University of Bamberg	Mendler
Nijmegen University		Geuvers
University of Białystok		Trybulec
Royal Holloway, Univ of London		Luo
	University of Birmingham	Ritter
	University of Kent	Thompson
	University of Nottingham	Altenkirch
	University of Sheffield	Fairtlough
	University of Toulouse	Soloviev
University of Edinburgh		Pollack
University of Manchester		Aczel
	University of Swansea	Setzer
University of Torino		Berardi
	University of Novi Sad	Ghilezan
University of Udine		Honsell
	University of Padova	Sambin
University of Warsaw		Urzyczyn
	University of Krakow	Zaionc
IoC Tallinn		Uustalo

Figure 3: List of Sites and Subsites

A.2 The Chalmers site

The Programming Logic Group within the Computer Science Department at Chalmers University has participated in all the previous EU funded TYPES working groups and coordinated one of them.

The Programming Logic Group has long-standing experience and expertise in the area of Intuitionistic Type Theory and its application to programming and mechanisation of proofs. The proof assistants ALF and AGDA, with the user interface ALFA were developed by this group.

The group plans to contribute to three project objectives.

- In Proof Technology, Peter Dybjer, Patrik Jansson and Marcin Benke are working on type theory and generic programming. The group has been actively involved in starting the Cover project (Combining Verification Methods) — a collaboration between the three main research groups at the department. It is expected that the proof assistants mentioned above will play an important role in this project. Of particular interest are methods to combine testing and interactive program verificiation. The role of automatic theorem proving is also interesting.
- In Foundational Research, Thierry Coquand is collaborating with Randy Pollack from Edinburgh on type systems for dependent records. Ana Bove is interested in proof methods for termination checking. She has worked a few months at the Nijmegen site and she is still actively collaborating with Venanzio Capretta and interacting with Yves Bertot (both currently at the Sophia Antipolis site). Aarne Ranta and Bengt Nordström are currently using type theoretical ideas in the area of natural languages technology, in collaboration with the Linguistic Department at Gothenburg University.
- In Formal Mathematics, Thierry Coquand has started to interact with Bas Spitters, now employed in the Nijmegen team, on computational mathematics.

The senior members of the Programming Logic Group are

- Bengt Nordström, contact person
- Ana Bove
- Marcin Benke
- Catarina Coquand
- Thierry Coquand
- Peter Dybjer
- Patrik Jansson
- Aarne Ranta
- Jan Smith

The group organised an international Workshop on Termination and Type Theory in the fall 2002. Several members of the group have been teaching in the TYPES and APPSEM summer schools series. Last year three post docs from the LogiCal research group visiting the department. In the spring, Alexandre Miquel, from the LogiCal group in France, came to give a course on introduction to type theory.

Short CV for one key researcher: Bengt Nordström was born in 1949 and has been professor of Computing Science at Chalmers University of Technology since 1986. He obtained his PhD from University of Umeå in 1978. His main research interest is logic for programming and in particular type theory. He has written (together with two coathors) a standard text on Martin-Löf's type theory.

Nordström's current interest is in document structure and editors for proofs, programs and documents. He is editor for *Nordic Journal of Computing* and *Journal for Universal Computer Science*. He has been involved in a number of European projects (Types, Clics and Appsem).

A.2.1 The Bergen subsite

The subsite Bergen is part of the Programming Theory Group at the Institute for Informatics of the University in Bergen.

The members participating are:

- Marc Bezem, contact person
- Hoang Anh Truong

The group is mainly going to contribute to the project objective Correctness of Computer Systems. Bezem and Truong are working on typing systems for component software in the framework of the project MoSIS (Modularity in large Software and Information Systems, Norwegian Science Council.) Other research interests are formal verification and automated proof search.

The following projects are collaborations with other members of the TYPES community.

- A chapter in the book on Typed Lambda Calculi edited by Barendregt (Nijmegen site).
- The book Term rewriting Systems (Cambridge University Press, ISBN 0521391156) co-authored and co-edited by Bezem and, among others, Klop (Nijmegen site).
- A joint research proposal called Automating Geometric Logic by Bezem and Coquand (Gothenburg site).

A.2.2 The Helsinki subsite

The group in Helsinki is known for its work on the application of type theory to natural language (Aarne Ranta, now at Göteborg) and to constructive geometry (Jan von Plato). Both applications have led to extensive subsequent work in several sites of the Types group, including computer implementation with different proof editors.

The group will mainly contribute to the project objective Foundational Research. The group is interested in the application of methods from proof theory to decision problems in formalized mathematics. This includes lattice theory and elementary geometry, and the decision problem of intuitionistic modal logic - a problem that had resisted attempts at a solution for several decades but now seems to be solved. This last topic is of considerable interest for computer science applications of logic.

The senior members are:

- Jan von Plato , contact person
- Petri Mäenpää, Nokia
- Sara Negri

The area in which most collaboration is expected is in the development of proof systems for constructive geometry on which at least two other sites have worked or are presently working (INRIA Sophia-Antipolis, Nancy). The earlier collaboration with Göteborg (Aarne Ranta on the proof editor PESCA and Thierry Coquand on the proof theory of order relations) is planned to be continued.

A.2.3 The Stockholm–Uppsala subsite

The logic groups of the departments of Mathematics in the universities of Stockholm and Uppsala have long experience in the fields of Type Theory, as well as in related fields like Category Theory, Topos Theory, Domain Theory, and Recursion Theory. Per Martin-Löf, the inventor of Constructive (or Intuitionistic) Type Theory, is the leader of the logic group in Stockholm. Constructive Type Theory is the major source of inspiration for most type theories under consideration in the TYPES network. The proof assistants ALF and AGDA, developed at the Chalmers University site, is based on Martin-Löf's type theory.

The group plans to contribute mainly to the Foundational Research project objective. This includes constructive and predicative development and formalization of mathematics; in particular Topology, Algebra, and Category Theory. The use of type-theoretical universes in such developments will be investigated.

The senior members of the logic groups are

- Per Martin-Löf, contact person
- Erik Palmgren
- Viggo Stoltenberg-Hanssen

Per Martin-Löf has given several invited talks in the TYPES workshops during the last couple of years. The group also had several visitors from other TYPES sites, e.g. Thierry Coquand, from the Chalmers site, and Nicola Gambino, from the Manchester site.

The group would like to cooperate in the near future with the Nijmegen site with the aim of mutually learning more about benefits and drawbacks of the different type theories in use.

A.3 The CNRS / Paris 7 site

The "Proofs, Programs, Systems" team of Univ. Paris 7 is a research team straddling the Mathematics and Computer Science departments. It participated to all the previous TYPES projects.

The team has a long-standing experience in mathematical logic, λ-calculus, proof theory and applications to programming. Main interests related to the TYPES project are in the design and development of proof assistants and in the study of computational content of logical systems and mathematical proofs.

The senior members of the "Proofs, Programs, Systems" team are:

- Michel Parigot, contact person
- Patrick Baillot
- Chantal Berline
- Pierre-Louis Curien
- Vincent Danos
- Roberto Di Cosmo
- Thierry Joly
- Delia Kesner
- Jean-Louis Krivine
- Yves Legrandgérard
- Pascal Manoury
- Alexandre Miquel
- Virgil Mogbil
- Paul Rozière
- Marianne Simonot

The team has numerous exchanges with other sites of the project. A. Miquel is coming from INRIA LogiCal project and visited Chalmers University during several months. T. Joly spent a year in Nijmegen. R. Matthes from LMU Muenchen spent six months in our team. C. Berline is collaborating with S. Berardi (Torino site), Patrick Baillot and Virgil Mogbil with Claudia Faggian (Padova site), P.L. Curien with H. Herbelin (INRIA site), etc.

Y. Legrandgérard, P. Manoury, M. Parigot, P. Rozière and M. Simonot are working on the design of a proof assistant, MathOS. P. Rozière is also participating in the development of the Phox system from the Chambery subsite.

J.L. Krivine, P.L. Curien, M. Parigot and A. Miquel are studying the computational content of proofs. M. Parigot and P.L. Curien work on distinguished properties of the classical Curry-Howard isomorphism, mainly duality and non determinism. J.L. Krivine is developing typed lambda-calculus in classical Zermelo-Fraenkel set theory with foundation and dependent choice and investigating the computational content of the full axiom of choice and of large cardinal axioms. A. Miquel works on the denotational semantics of type theories and on the computational contents of the proofs of (intuitionisitic) set theory.

P. Baillot, V. Mogbil, C. Berline, T. Joly, R. Di Cosmo and D. Kesner are developing specific aspects of the syntax and semantics of type systems. P. Baillot and V. Mogbil are working on the use of type systems and proof-theoretical methods for polynomial time computation. C. Berline is working on semantical aspects of type systems: System F and Map Theory. T. Joly is working on the fine structure of typed λ-calculus. R. Di Cosmo is working on computable type isomorphisms. D. Kesner is working on typed lambda-calculus with explicit substitutions.

Vincent Danos is working on the formalization of a proof of the four colours theorem.

Short CV for one key researcher: Michel Parigot is Researcher in Mathematics at CNRS (National Center for Scientific Research) since 1982. He got his PhD in Mathematics in 1980 at University Paris 7. He worked in several fields of Mathematical Logic: he started with Model Theory and then turned to Proof Theory, Lambda-Calculus and applications to Programming through the computational interpretations of proof systems. He introduced in 1992 a computational interpretation of classical logic, the lambda-mu-calculus, which has been widely studied and used during the past decade.

A.3.1 The Université de Savoie subsite

The Logic team of the math laboratory within the Université de Savoie has participated in all the previous EU funded TYPES working groups as a subteam of Paris VII team.

We have worked with pure and typed lambda-calculus and C. Raffalli is the author of the PhoX proof assistant, a proof assistant which is very easy to learn (this is a design requirement of the system) and used for teaching math.

Current interests of the team include proof assistants, proof and extraction of programs using classical logic, process calculi, proof of normalisation of typed calculi.

The senior members of the Programming Logic Group are

- Christophe Raffalli, contact person
- René David
- Karim Nour
- Noël Bernard
- Jacques Doyen

We have deep contacts with the Paris VII team arround the implementation of PhoX.

C. Raffalli and R. David have a project with the INRIA Nancy and the linguistic team of Paris VII to develop an interface using natural languages for PhoX (this project is founded by an ACI).

C. Raffalli, R. David and P. Roziere work on the teaching application of PhoX.

C. Raffalli works an a process calculi with Curry Howard. R. David and K. Nour work on proof of strong normalisation for various classical proof system.

A.4 The INRIA-Sophia site

The large and active group at INRIA Sophia-Antipolis has been part of all previous EU funded TYPES projects. The group has made many large formal proof developments using the Coq proof tool, including formalisation of programming language semantics (e.g. Java, in collaboration with other TYPES teams in Paris Sud and Munich), and formalization of mathematics (e.g. formalization of rational numbers, and of Gröbner bases, in collaboration with Nijmegen and Chalmers). INRIA Sophia and INRIA Rocquencourt participate in the INRIA action "Concert: Certified Compiler"; this is a broad project on the certified implementation of compilers for "real" programming languages such as C. Members of the team have done much foundational work on induction reasoning with higher-order specifications. The team has also been active in building interfaces for proof tools. Yves Bertot (with Pierre Castéran of University of Bordeaux) has written a textbook about using the Coq proof system for practical applications.

The senior members are

- Joëlle Despeyroux, contact person
- Gilles Barthe
- Yves Bertot
- Marieke Huisman
- Luigi Liquori
- Loïc Pottier
- Laurence Rideau
- Laurent Théry

The site plans to contribute to the following objectives of the current proposal.

Correctness of Computer Systems The group is interested in programming with dependent types, and also plans to work on some challenge for practical formalization, such as computer arithmetic.

Foundational Research The group is actively involved in work on inductive definitions, subtyping, rewriting, termination, logical frameworks and type isomorphism.

Formal mathematics The group will do case studies, and particularly participate in building a repository of formal mathematics.

Short CV for one key researcher: Joelle Despeyroux is a researcher at INRIA since 1982. Her main research interests lies in the area of formal methods and logical frameworks. Her main contributions in these domains are the proposition of "natural semantics" and new types theories to improve both the formal specifications and the mechanization of proofs of properties of languages such as the correctness of a compiler.

A.4.1 The INRIA-Futurs site

The LogiCal project of INRIA-Rocquencourt has moved to INRIA-Futurs at the École polytechnique. The main topic is the development of proof systems, in particular the Coq system, but it is also involved in the development of formal libraries and challenging formal proofs and in the study of logical formalisms such as type theory, inductive types, sequent calculus and deduction modulo.

The senior members of the group are

- Gilles Dowek, contact person
- Bruno Barras
- Hugo Herbelin
- Jean-Pierre Jouannaud
- Dale Miller
- Benjamin Werner

The group has a strong cooperation with the Paris Sud group. We have strong cooperation with two industrial subsites : Trusted Logic and France Telecom. We have exchanged doctoral and post-doctoral students with Chalmers, Warsaw, Cambridge, Sophia. We have frequent cooperation with most of the sites of the proposal.

Members of the group have been teaching in the various Types summer schools.

Short CV for one key researcher: Gilles Dowek. Scientific leader of the LogiCal project at INRIA (2000-). Professor at the École polytechnique (2003-), Researcher in residence in the laboratory ICASE-NASA Langley and the *National Institute of Aerospace* at Hampton, U.S.A. (2000-). Formerly Research Scientist at the Institut National de Recherche en Informatique et en Automatique (INRIA). Member of the program committee of several conferences (LICS, TLCA, CADE, RTA, LPAR, CSL, TPHOLs, ...), CADE Trustee. Member of the steering committee of the european project Types (1999-2003). Grand Award at the Prix D'Alembert des Lycéens for a conference for high school students. (2000) Award at the *European Philips Contest for Young Researchers and Inventors (1983)*. Several popular science books published.

A.4.2 The Bologna subsite

The Helm group (Hypertextual Electronic Library of Mathematics) is active in Bologna since 1999. The work of the group has been mainly focused on the application of information technologies (e.g. XML) to the problems of management and (web) publishing of large repositories of formal mathematical knoweldge. We are coordinating the European FET-open Project IST-2001-33562 MoWGLI (http://mowgli.cs.unibo.it/), which includes the TYPES sites at Sophia-Antipolis, Nijmegen and Trusted Logic. We recently organized and chaired the Second International Conference on Mathemtical Knowledge Management, in Bertinoro Italy (February 2003). We are also developing our own proof assistant (a clone of Coq), where the several components are all centered around and communicate through the XML low-level encoding of the mathematical entities.

Our main contribution to the TYPES project will be in the area of formalisation of mathematics, focussing on intefaces and tools for the automation of formal reasoning, and the management "in the large" of mathematical information. Our work is integrated with the Coq proof tool, and we plan to apply our approach with other TYPES proof tools.

The senior members of the Helm Group are

- Andrea Asperti, contact person

- Luca Padovani
- Irene Schena
- Ferruccio Guidi

A.4.3 The Dassault Aviation subsite

Dassault Aviation has put formal methods into industrial practice, especially formal specification and verification of safety critical embedded software. Our group investigated theorem proving techniques based on typed lambda calculi, using the Coq proof tool. (This choice was driven by tool-qualification issues, since correctness of software verification tools is a key issue for certification by Airworthiness Authorities.) Dassault Aviation funded a PhD student (Antonia Baala) in the Lemme Project at INRIA Sophia to improve the handling of non structural recursive functions in the Coq System (defended in 2002).

We are now using the Coq system to prove assertions on safety critical embedded C programs using the Why proof-obligations generator from Paris Sud. We intend to use Why extensively in the fall of 2003 and in 2004.

The senior members of the Programming Logic Group are

- Emmanuel Ledinot, contact person
- Dillon Pariente
- Claire Campan

A.4.4 The Minho subsite

The group at the University of Minho, Braga, Portugal, was involved in the previous TYPES project. The researchers work on constructive subtyping, extensible definitions, type-based termination and the study of computational content of proof systems. There is extensive collaboration with TYPES sites at INRIA Sophia-Antipolis and Tallin

In this proposal, the Minho site will mainly work on the objective of foundational research, particularly the areas of subtyping and termination.

The key members involved are:

- Luís Pinto (contact person)
- José Espírito Santo
- Maria João Frade

A.5 The Université Paris-Sud site

The team at Université Paris Sud in Orsay, France has recognised skills in the development of interactive proof assistants (Coq) but also automatic proof tools based on rewriting (Cime). It has developed specialised tools (Why, Krakatoa) on top of Coq for verification of programs written in C or Java. It is also recognised for its contribution to dependent type theory, in particular modules, combination with rewriting, inductive types and program extraction.

The team will mainly contribute to the objectives of Correctness of Computer Systems and Proof Technology.

- We shall continue our development of environments based on Type Theory dedicated to a specific verification area (mathematics, Java or C programs, automata). It involves to design in type theory good models for the representation of the domain of study, to specialize proof strategies and provides appropriate interface. (Collaboration with Nijmegen, INRIA Sophia-Antipolis, France Telecom, Grenoble).
- We are contributing to the effort to design a usable programming language based on Dependent Type Theory (Collaboration with Chalmers, Durham, Edinburgh and Nottingham).
- We are contributing to the development of the Coq system (Collaboration with INRIA Futurs and the Coq-user sites).
- We are studying the interaction between interactive and automatic proofs in the domain of rewriting and proof of termination (Collaboration with INRIA Futurs, France Telecom).

The senior members are

- Christine Paulin-Mohring, contact person
- Sylvain Conchon
- Evelyne Contejean
- Judicaël Courant
- Jean Duprat (ENS Lyon)
- Jean-Christophe Filliâtre
- Claude Marché

Our research on development of certified Java programs is funded by the IST VERIFICARD project and the national ACI Geccoo. The national project RNTL AVERROES (analysis and verification for the reliability of embedded systems) uses the Coq system for the modelling and verification of systems described by a general class of timed-automata.

Short CV for one key researcher: Christine Paulin is working at the Laboratoire de Recherche en Informatique (LRI) as a full professor from the univerity Paris Sud in Orsay since 1997. From 1989 to 1997, she had a permanent research position from CNRS and was working at the Laboratoire de l'Informatique du Parallélisme (LIP) at Ecole Normale Supérieure in Lyon.

C. Paulin has been leading the INRIA Coq project from 1997 to 2000 and is now leading the Démons team at LRI and co-leading with Gilles Dowek the LogiCal project which involves INRIA, Ecole Polytechnique and University Paris Sud.

Her research concerns the theory and practice of proof assistants based on Type Theory, and especially the Coq tool. Her main concerns are the application of these tools to program certification. Her contributions are in the area of program extraction from proofs, inductive definitions and modelisation of programs (timed automata, memory models for Java programs).

A.5.1 The Grenoble subsite

The Verimag laboratory at the University Joseph Fourier is a leading research center in embedded systems, with long standing experiences in the development of formal verification tools. It will mainly contribute to the objective of Correctness of Computer Systems through its research activities in temporal logics, concurrency theory and proof assistants.

The contact person is Jean-François Monin. He was head of the formal methods group at France Telecom R&D until summer 2003, and has been in the TYPES community since the 1990's. He worked on typing and correctness proofs of functional programs with exceptions, and on the verification of telecommunication protocols in a type theoretical framework. His current research interests include programming with dependent types, and the formalization/verification of quantitative properties of programs, with applications to the security of software systems.

A.5.2 The France Telecom R&D subsite

The Formal Methods Group in FT R&D participated in the previous EU funded TYPES working group. It has long experience in using logic based formalisms, including type theory, in a number of telecommunication applications: communication protocols, cryptographic protocols, embedded platforms. It has experience using various proof systems and verification tools, e.g. HOL, PVS, Coq, B-tools, SMV and Kronos.

The group will mainly contribute to the objective of Correctness of Computer Systems through its work on programming with dependent types, static analysis, deduction systems and interfaces between them. Some application fields which are considered are modelling communicating systems over cellular networks and verification of security properties.

The Formal Methods Group contributed actively to the development of the Coq system through a number of contributed theories, tactics (Presburger arithmetic, binary arithmetic, rewriting) and other tools, such as the experimental Coq-ELAN interface. The group has sustained relationship with national projects LOGICAL, PROTHEO, LEMME and LANDE at INRIA, and the Demons group at LRI-Paris Sud. It was also an end-user panel member of the Verificard IST.

The senior members of the group involved in TYPES are

- Cuihtlauac Alvarado, contact person
- Pierre Cregut

A.6 The LMU München site

This site includes mathematics and computer science groups at Ludwig-Maximilians-Universität, München. The site is very strong in logic and foundations, especially proof theory (and its application to computer programs), semantics of dependent types, and feasible and resource-limited computation. The MINLOG interactive proof system has been developed by this site. The site has been a part of several previous EU TYPES projects, and has a joint PhD programme “Logic in Computer Science” with the TU München site (financed by the German Research Foundation).

The site plans to contribute to the objectives of this proposal mainly in the following areas.

Correctness of Computer Systems The site collaborates with Edinburgh on Mobile Resource Guarantees. There is work on justification of programming language concepts inspired from proof theory, such as continuations, and on specification and verification of functional programs with higher-order inductive and coinductive datatypes.

Formal Mathematics and Mathematics Education Work on formal topology within type theory, and on mathematics education using the Minlog Proof Checker.

Foundational Research Ongoing work includes type based termination, program development by proof transformation, program extraction from classical proofs and program extraction by realizability interpretation. Also of interest are intensional behaviour of datatypes, expressiveness of fragments of polymorphic lambda calculus and representation of binding syntax by nested datatypes.

The key members involved are

- Helmut Schwichtenberg, contact person
- Andreas Abel
- Klaus Aehlig
- Wilfried Buchholz
- Martin Hofmann
- Hans-Wolfgang Loidl
- Ralph Matthes
- Peter Schuster
- Olha Shkaravska

Short CV for one key researcher: Helmut Schwichtenberg, born 1942. 1968 Dr.rer.nat., Universität Münster. (Adviser: D. Rödding). 1971/72 Research at Stanford University (with Feferman, Kreisel). 1974 Habilitation für Mathematik, Universität Münster. 1974-1978 Wiss. Rat und Professor, Universität Heidelberg. 1978-present Professor (Ordinarius), Mathematisches Institut, Universität München. 1981/82 Research at Stanford University (with Feferman, Sieg). 1986 Member, Bayerische Akademie der Wissenschaften. 1986/87 Research at Carnegie-Mellon-University (Pittsburgh). Fall 2000 Visiting Professor at Stanford University.

A.7 The TU Munich site

The group at TU Munich, which has participated in the last two TYPES projects, is focused on the development of both further improvement of the theorem prover *Isabelle*, and on its applications. On the improvement side, our group has developed the *Isar* environment for writing human-readable proof documents, as well as a module system called *Locales*, which allows for a more elegant and scalable presentation of mathematical theories. This work led to a joint journal paper with the Nijmegen group comparing the Isar and Mizar systems.

Moreover, thanks to a recently developed extension of its kernel, Isabelle can now generate primitive proofs, represented as λ-terms in the spirit of the *Curry-Howard isomorphism*. This allows for an independent verification of proofs produced by Isabelle and forms the basis for future applications such as proof-carrying code, or the exchange of proofs with other theorem proving systems. Based on this representation of proofs, we have introduced a new generic framework for the extraction of programs from constructive proofs conducted in Isabelle. This has been carried out in cooperation with the group of Prof. Schwichtenberg from the LMU Munich site.

On the applications side, emphasis was on programming language semantics, and formalised mathematics. We have continued our work on formalising programming languages and treated substantial parts of the Java language, such as operational semantics for both the source and bytecode level, a compiler from source code to bytecode, as well as Hoare logics. We are in the process of obtaining the first bytecode verifier for the Java Virtual Machine that has a machine-checked correctness proof, covering even advanced features such as bytecode subroutines and object initialization. Moreover, we are currently evaluating both the concept and the implementation of Locales by applying it to a formalisation of abstract algebra. An Algebraic Base Library for Isabelle, of which a first version was released with Isabelle 2003, is being developed jointly with L. Paulson (University of Cambridge), and H. Kobayashi (Nihon University). The developed algebraic library will be used to prove results in algorithmic algebraic topology, which is a first step to increase the reliability of symbolic computation systems in this area.

Future research will mainly focus on the objective of Correctness of Systems:

Integrated environment for proving and programming A long term goal is to achieve a tighter integration of specification, proving, programming and testing. To this end, we have already extended Isabelle with a mechanism for generating executable code from specifications, which covers inductive datatypes, inductive predicates and recursive functions. Based on this code generator, we are currently developing a framework for testing specifications by evaluating logical formulae with respect to a mapping of free variables to random values. This project is inspired by the QuickCheck tool developed at the Chalmers site, which was originally designed for testing Haskell programs, and has now also been integrated into the Agda / Alfa proof editor.

Proof-carrying code We are currently developing a formally verified framework for proof-carrying code. A particular application of this framework is the certification of resource bounds (i.e. for time and space) of programs. In this respect, we will closely collaborate with research groups at LMU München (Prof. Martin Hofmann) and Edinburgh (Dr. David Aspinall).

The senior members involved in the TYPES project are:

- Tobias Nipkow (contact person)
- Clemens Ballarin
- Stefan Berghofer
- Gerwin Klein
- Martin Strecker.

Short CV for one key researcher: Stefan Berghofer studied computer science at the Technische Universitaet Muenchen from 10/94 to 5/99, his diploma thesis was on the definitional construction of inductive data types in Isabelle/HOL. Under the supervision of Tobias Nipkow, he then worked on his PhD thesis entitled "Proofs, Programs and Executable Specifications in Higher Order Logic". This work, which was about program extraction from constructive proofs in Isabelle/HOL, was completed in 10/03. Since 10/03 he works as a postdoctoral researcher at the Technische Universitaet Muenchen. His research interests include the design and implementation of theorem provers, inductive definitions and constructive logic.

A.7.1 The Bamberg subsite

The Informatics Theory Group at The University of Bamberg has strength in the applications of intuitionistic modal logic and modal type theory, focussing on the formalisation and validation of abstraction and refinement and the associated compositionality problem. The group has long-standing collaboration with the Sheffield subsite, and members have contributed to previous Types meetings and proceedings, organised workshops on intuitionistic modal logic, and edited special journal issues on this topic.

The group plans to contribute to the objectives of this proposal mainly in the area of foundational research on the use of type theory for coherent interface models in synchronous component-based programming. This work now provides a new common research ground with researchers at the INRIA Sophia-Antipolis site. The group recently has begun collaborations with Birmingham on the constructive semantics of CS4 modal logic. The group plan to exchange visits with the the Padova site which shares similar research interests.

The senior members of the Bamberg subsite are

- Michael Mendler (contact person)
- Joaquín Aguado

A.8 The Nijmegen site

The Foundations group of the Computer Science Department has a long history and expertise in (1) lambda calculus, (2) type theory and its relation to logic, and (3) formalizing mathematics, especially in type theory based systems, but also in more general. The group has completed a full formalization of (a constructive proof of) the Fundamental Theorem of Algebra (FTA) and the Fundamental Theorem of Calculus (FTC) in Coq. This formalization is continuously extended and updated, thus creating a large mathematical repository. The groups members are knowledgeable Coq users but also have expertise in Mizar and HOL-light. Furthermore the group has a great expertise in constructive logic and mathematics.

The team will mainly contribute to the objective of Formal Mathematics and Mathematics Education and to a lesser extent to Proof Technology and Foundational Studies. Currently, the group's main focuses are

- Mathematical libraries and repositories. (Collaboration with INRIA Sophia-Antipolis, INRIA-Futurs, Orsay, Bologna, Białystok.)
- Formalizing constructive and exact analysis as a case study, but also in relation to the computational aspects of mathematics. (Collaboration with Göteborg, INRIA Sophia-Antipolis)
- Closing the gap between formal and informal mathematical proofs, (Collaboration with Białystok, Bologna, INRIA Sophia-Antipolis)
- Proof presentation: how to present a formalized proof in an (interactive) document, e.g. on the web.(Collaboration with Bologna, INRIA Sophia-Antipolis)
- Interactive theorem proving: how to (formally) model the interaction between a mathematician and a theorem proving system and how to present an unfinished proof in an understandable way and to interact with it. (Collaboration with Białystok, Bologna, INRIA Sophia-Antipolis, INRIA Futurs)
- Write an overview book on typed lambda calculus. (Collaboration with Torino, Udine, INRIA Futurs, Bergen)

The senior members of the Foundations group are

- Herman Geuvers, contact person
- Henk Barendregt
- Jan Willem Klop
- Rob Nederpelt from TU Eindhoven
- Wil Dekkers
- Freek Wiedijk
- Bas Spitters

Our research on Formalizing Mathematics is partially funded by a national "Spinoza" grant of Barendregt for outstanding researchers. The research on proof presentation and closing the gap between formal and informal mathematics is partially funded by the Calculemus EC-TMR and partially by the IST-FET project MoWGLI.

Short CV for one key researcher: Dr. H. Geuvers is associate professor in the Foundations of Computer Science group of the University of Nijmegen. He studies mathematics and did his PhD in Nijmegen under supervision of prof. Barendregt in 1993 on Logics and Type Systems. After that he worked for a number of year at the university of Eindhoven, before returning to Nijmegen. Geuvers has worked a lot on the meta theoretic aspects of types systems and on developing type systems for formalizing mathematics. In this field his work has been very influential. recently he has been the driving forceafter large mathematics formalization projects in Nijmegen: the FTA project and the C-CoRN project. Geuvers has been reviewer for many PhD theses in the area of type theory and he is the Nijmegen reponsible for the European projects Mowgli and Calculemus that both focus on computer formalization of mathematics.

A.9 The Białystok site

The Mizar Group http://mizar.org (now at the University in Białystok - UWB) led by Andrzej Trybulec has been developing a proof checker system for almost three decades. The input language of the system, called Mizar, is close to informal mathematical language and enjoys rich linguistic structure: open vocabularies, flexible syntax, dependent types (allowing for overloading and hidden arguments), adjectives and structures (record types) with inheritance. The system is based on classical logic - Fitch-Jaśkowski system of conditional proofs.

The main activities of the project are:

- evolving the language itself, and software for processing, particularly verifying the correctness of, text written in Mizar, called Mizar articles
- development of the Mizar Mathematical Library (MML) the largest repository of computer checked mathematics in the world. As of April 2003, Mizar Mathematical Library consists of 774 articles authored by ca. 130 persons from 10 countries (40 000 theorems). The articles submitted to MML, after automatic translation to English, are published in *Formalized Mathematics*; the electronic version is available at http://mizar.org/JFM.

UWB cooperates with KUN (Nijmegen). The cooperation includes parallel development, e.g. the formalization of a proof of FTA done almost simultaneously with some theoretical work. We continue long term and fruitful exchange of the ideas with U. of Durham (mathematical vernacular, subtyping) and other groups.

In this proposal, the group will mainly address the objective of Formal Mathematics, focussing on mathematical vernacular language and the use of ideas from type theory, such as coercive subtyping or generic metastructures (modules in Coq or locale in Isabelle), in classical mathematics.

The senior members of the Mizar Group are

- Andrzej Trybulec, contact person
- Grzegorz Bancerek
- Czeslaw Bylinski
- Adam Grabowski
- Robert Milewski
- Adam Naumowicz
- Krzysztof Retel

Short CV for one key researcher: Andrzej Trybulec graduated from Warsaw University in 1966. In 1974 he received Ph. D. in mathematics from the Institute of Mathematics, Polish Academy of Science. The thesis was dedicated to shape theory (a generalization of homotopy theory). Since 1974 he has led the Mizar project. His recent publications and scientific interests concern interpretations of the positive part of intuitionistic propositional calculus and dependent types algebras. He is an active member of EU working groups CALCULEMUS and MKMNET. For a long time he has collaborated with TYPES, formally since 1999. Currently, he is Deputy Director for Science in the Institute of Computer Science, University of Bialystok.

A.10 The Royal Holloway site

The Department of Computer Science at Royal Holloway, University of London has substantial expertise in type theory and its applications to theorem proving and programming. The group was in the University of Durham and was the co-ordinating site of the previous TYPES consortium. In the last several years, it has made contributions to TYPES-related research topics, including development of logical frameworks and type theories, development and implementation of coercive subtyping, implementation of proof assistants, development of proof technology, and design of a platform for dependently-typed programming.

The activities will contribute to the objectives of the proposal:

- Correctness of computer systems: We will work on dependently typed programming, among others. Currently, a 3-year project funded by UK EPSRC is in progress to develop the basic technology and a platform for dependently-typed programming. There is a planned workshop on this topic (in year 1) in the current proposal.
- Formal mathematics and mathematics education: We will work on this topic by developing associated proof tools for mathematics, among other things, in the 3-year project Pythagoras, funded by UK EPSRC and joint with the Manchester site of this proposal.
- Proof technology: We will further develop the proof assistant Plastic, studying the implementation techniques and developing its applications (including domain-specific reasoning).
- Foundational research: We will continue to work on logical frameworks, subtyping, and extensionality.

The key members are:

- Zhaohui Luo, contact person
- Conor McBride
- Yong Luo

The work of the group is currently supported by two UK EPSRC grants, as well as the EU TYPES grant. It has close collaborations with several TYPES sites and has had fruitful visits from and to other TYPES sites, including Manchester, Paris Sud, Nottingham, and Edinburgh. These collaborations and visits have greatly advanced the research activities; the TYPES consortium provides an excellent platform for the research activities in this area.

Short CV for one key researcher: Zhaohui Luo is Professor of Computer Science at Royal Holloway, University of London. His research has focused on type theory and its applications to computer-assisted formal reasoning and functional programming. Luo has been the coordinator of the previous EU Thematic Network TYPES. He is the principal investigator of the projects, funded by the U.K. research council EPSRC, on programming with dependent types (GR/R72259, Dec. 2001 - Apr. 2005) and on machine support of proof-oriented mathematics (GR/R84092, Apr. 2003 - Mar. 2006). His publications include two books on type theory and programming methodology.

A.10.1 The Birmingham subsite

The Birmingham group is in the Computer Science Department of the University of Birmingham. The group is working on both foundational and practical aspects, including work on semantics for proof search, program verification using linear logic to model state change, probabilistic model checking, automated verification for distributed systems, and representations for mathematics. Research in the group is supported by EPSRC, ESPRIT and the Midlands e-Science Centre.

The main contributions of the group to the objectives will be:

- Correctness of Computer Systems: to use linear logic to describe state change in object-oriented languages;
- Formal mathematics and mathematics education: to continue the work on representation for mathematics, developing an object-centred and hierarchical representation of concepts by frames that approximates mathematical practice;
- Proof technology: to work on automated verification techniques, including combining model checking and theorem proving;
- Foundational research: to work on a semantics describing control during proof search.

The senior members of the group are:

- Eike Ritter, contact person
- Achim Jung
- Manfred Kerber
- Marta Kwiatkowska
- Uday Reddy
- Mark Ryan

In the previous TYPES project, Ritter has visited Padova several times, in connection with the work on program verification. The group is also collaborating with Nottingham in teaching courses about type theory to starting PhD-students.

A.10.2 The Kent subsite

The Functional Programming Group at the University of Kent has a long history of involvement with dependdent type theories. Thompson is the author of an early (1990) introduction to the area and Hanna is the implementor of a dependently-typed system, Veritas, used in early hardware verification work. Current interests of the group centre around building systems to make functional programming systems more usable. These include work on refactoring in FP languages such as Haskell, foundations of functional programming type systems, and Computer algebra and verification.

The group will make contributions to the following objectives:

- Correctness of computer programs: to work on functional programming and dependent types.
- Formal mathematics and mathematics education: to continue work in verified computational mathematics with Therese Hardin's group at LIP6, Paris, who are, in turn, collaborators with the Coq group, the Paris Sud and INRIA sites of the current proposal.

Dependently typed systems provide a major challenge for implementors and system designers alike. The group also aims to extend the projects in refactoring and visual programming to embrace dependent types, in collaboration with groups at Gothenburg and elsewhere. This will also link with the Cover project at Gothenburg.

The senior members of the Functional Programming Group are

- Simon Thompson, contact person
- Keith Hanna
- Stefan Kahrs
- Claus Reinke

A.10.3 The Nottingham subsite

The Foundations of Programming group is at the Department of Computer Science of the University of Nottingham. The aim of the group is to make substantial gains in the effectiveness of mathematical reasoning and apply such gains to the construction of computer programs. The research of the group is currently structured around the following main themes: Algebra of Programming, Functional Programming, and Logic in Computer Science. The group has its own series of research meetings, and is a founder member of the Midlands Graduate School in the Foundations of Computer Science, which provides postgraduate courses for PhD students in theoretical computer science. Research in the group has been supported by grants from EPSRC, ESPRIT, Microsoft Research, the British Council, and the (Dutch) Nederlandse Wetenschappelijke Organisatie.

The group will mainly contribute to the following objective:

- Correctness of computer programs: Hutton and Altenkirch are currently colloborating with McBride and McKinna from the Durham site on the design and application of dependently typed programming langauges.

The senior members of the Foundations of Programming Group are

- Thorsten Altenkirch, contact person
- Natasha Alechina
- Roland Backhouse
- Louise Dennis
- Graham Hutton

Altenkirch has actively participated in the previous Types Working Group as a member of the Munich site and has regularly contributed to the annual Types workshops. Dennis is investigating proof planning using technologies developed in the Types community with potential applications to the design of new proof editors.

A.10.4 The Sheffield subsite

Fairtlough and Barry Norton are part of the Verification and Testing group in the Department of Computer Science at Sheffield University. This group also includes Graham Birtwistle as Professor Emeritus and has links with Prof. Michael Mendler's Lehrstuhl "Grundlagen der Informatik" at the University of Bamberg (Bamberg subsite). Fairtlough works with Mendler on a logical approach to computational monads which is known as Lax Logic; this work integrates techniques more commonly used in functional programming with formal verification in Type Theory and in other logical frameworks. The primary application of Lax Logic is to provide a formal framework within which systems can be described and verified over multiple abstraction levels. This work has been recently supported by EPSRC. Norton has been carrying out research with Mendler within the EPSRC-funded RealType project whose aim was to develop a theory of reactive types for software components used in the development of signal processing applications. His work is in developing techniques which integrate process-algebraic and type theoretic approaches to handling components. Birtwistle has made major contributions to the formal verification of hardware using both process algebra and higher-order logic and continues to work in this area.

The group will contribute to the following objective:

- Correctness of computer systems: to work on real-time systems and the associated approaches to logical reasoning.

The senior people involved are

- Matt Fairtlough, contact person
- Graham Birtwistle
- Barry Norton

A.10.5 The Toulouse subsite

The group Zeno (http://www.irit.fr/zeno/) is at IRIT, Toulouse. The types-related research topics include isomorphism of types, extensions of reductions systems with weak extensionality, coercive subtyping, and applications of proof assistants such as Coq. The Zeno group has organised the first international workshop on Isomorphism of Types (WIT2002) has had active collaborations with the Durham site.

The group will mainly contribute to the following objective:

- Foundational research: to work on type isomorphism and coercive subtyping.

The senior members of the group are

- Sergei Soloviev, contact person
- Prof Jean-Paul Bodeveix
- Dr Mamoun Filali-Amine (researcher CNRS)

A.11 The Edinburgh site

The Laboratory for Foundations of Computer Science (LFCS) in the School of Informatics of Edinburgh University, has participated in all four previous EU funded TYPES projects. LFCS workers developed the LEGO proof tool and the Proof General user interface, both widely used by researchers at other TYPES sites. LFCS researchers also contributed much to the theoretical basis on which all TYPES work rests, including topics such as logical frameworks, abstract syntax and variable binding, inductive definitions, consistency proofs and extensionality. This work is represented by many talks at TYPES workshops and publications in TYPES proceedings and other international refereed forums, often in collaboration with workers from other TYPES sites.

The site plans to contribute mainly to three project objectives.

- In proof technology, the future direction for the Proof General generic user interface for proof tools,[4] laid out in a white paper, is to re-engineer and expand the system, centering around a well-specified *protocol for interactive proof* to be implemented by message passing in XML.
- In foundational research, it is planned to continue the collaboration with the Chalmers site on semantics and implementation of logical frameworks with subtyping and modules.
- On correctness of computer systems, Edinburgh has strong collaboration with the Durham and Nottingham sites on programming with dependent types. We are discussing collaboration on this topic with other sites. Also Edinburgh has collaboration with LMU Munich on *mobile resource guarantees*, proof-carrying code where the proofs guarantee the code will adhere to a resource specification.

The senior members are

- Robert Pollack, contact person
- David Aspinall
- Jacques Fleuriot
- Paul Jackson
- Gordon Plotkin

Short CV for one key researcher: Robert Pollack, Senior Research Fellow. BSc in mathematics (MIT, USA); MSc in mathematics (Northeastern Univ., USA); PhD in computer science (Edinburgh Univ., UK). Over 10 years industrial software experience. Over 15 years research in Type Theory, machine proof and formal methods. Involved in all previous EU funded TYPES projects. Interested in large scale formal development in type theory and higher order logic, modularized proof, libraries of formal mathematical knowledge, mathematical vernacular languages. Recent work on modules and subtyping in type theory, in collaboration with the Chalmers site. Recent work on program logic for functional programming.

[4] The Coq, Isabelle, Plastic and PhoX tools from other TYPES sites all use Proof General as their default interface.

A.12 The Manchester site

The Manchester site is the Formal Methods and Mathematical Foundations groups, in the Manchester University Computer Science Department. It has participated in all previous EU funded TYPES working groups. The site has considerable expertise in mathematical logic, category theory, automated deduction and machine checked mathematics. Current interests of the site include proof assistants for mathematicians and mathematics education, logical frameworks, constructive set theory and constructive topology.

The site will mainly contribute to the objectives of Formal Mathematics and Mathematics Education and Foundational Research.

- Manchester has a UK/EPSRC funded project (Pythagoras), jointly with the Durham site of this proposal, to work widely on aspects of Formal Mathematics and Mathematics Education
- Peter Aczel and others will continue to work on constructive set theory and its relationship with dependent type theory. In particular their work on constructive topology (in constructive set theory) will be developed in parallel with the work on formal topology (in type theory) of the Padua subsite. There is also a planned workshop on Formal Topology in the current proposal.
- Work on logical frameworks, arising from Robin Adams' PhD work will continue.

The senior members group relevent to our proposal, are

- Peter Aczel (contact person)
- David Rydeheard
- Andrea Schalk
- Andrei Voronkov

The site has regularly taken part in the previous TYPES annual meetings, small workshops and site visits. The site has particularly close collaboration with the Durham site, including joint UK-funding, and many visits by researchers and students.

Short CV for one key researcher: Peter Aczel is professor of Mathematical Logic and Computing Science and works in both the Mathematics and Computer Science Departments at Manchester University. He has over 45 publications in the fields of mathematical logic and theretical computer science, including a monograph on non-well-founded sets. He has been an editor of the Journal of Symbolic Logic and the Annals of Pure and Applied Logic and is on the editorial Board of the Cambridge Tracts in Theoretical Computer Science. He is a well-established member of the European Types community who is on the steering committee of the previous and the planned EU Types working groups.

A.12.1 The Swansea subsite

The Swansea subsite, at the Dept. of Computer Science, University of Wales, is particularly strong in mathematical logic and foundations of type theory, but also has interests in applications to programming languages. The site plans to contribute to three of the objective areas

- Foundational research includes work on inductive and coinductive definitions, especially induction-recursion, on proof theoretically strong extensions of type theory, and on totality and termination.

- Work on correctness of computer systems includes programming with dependent types (especially interactive programs), object-oriented programming in type theory, and extraction of programs from classical proofs.
- Work on proof technology will be case studies applying interactive theorem proving to algebraic specification languages.

The senior members of the Swansea subsite are:

- Anton Setzer (contact person)
- Ulrich Berger
- Markus Roggenbach

A.13 The Torino site

The senior members are

- Stefano Berardi, contact person
- Mario Coppo
- Mariangiola Dezani
- Elio Giovannetti
- Ugo de' Liguoro
- Ines Margaria
- Simona Ronchi
- Maddalena Zacchi

The group's research is focused on the study of (compile-time) techniques for program analysis, using essentially tools and techniques of constructive logics and type theory developed by the other sites of the Types projects.

The main goal is the development of systems for formal reasoning about programs and the study of their application to program transformation and optimization. Type inference systems for various kind of analysis (like useless-code, strictness, and totality) of higher-order functional programming languages have been developed and studied, and the research mainstream is now evolving toward languages with imperative and distributed features. Further perspectives are in the study of software validation and security.

A foundational interest of Torino group in the field of constructive logics is in the programming-with-proofs paradigm in the style of logical frameworks like Coq. The main concern is with methods to obtain actual programs out of formal proofs, both in intuitionistic and in classical logic.

As next step in our research, we plan to implement a prototype of a Useless-Code Elimination (UCE) for higher-order typed functional programs with let-polymorphism. The long-term goal is being able to optimize code extracted from proofs in the Coq system by the other sites of the project. Several related applications are expected from this work, for instance a typechecker for rank 2 Intersection Types for (a subset of) Caml, the language used in the implementation of Coq.

On the foundational side of type theory, we plan to develop a semantical constructive interpretation of non-recursive maps and Classical Arithmetic, relating the constructive content of classical proofs to parallel typed computations.

Short CV for one key researcher: Stefano Berardi was orn in 1962. Ph.d. student, from 1986 to 1989. His Ph.d. thesis is quoted in the "Handbook of Logic in Computer Science" as a basic contribution toward a systematic understanding of typed lambda calculi. Member of the Projects "Types for Proofs and Programs" (1992-2003). Organizer and Editor, with M. Coppo, of the proceedings of the two congresses of such project (Turin, 1995 and 2003). Full Professor in C.S. Dept. of Turin University since 2001. He works in the Semantic and Logic of Computation group in such department, on typed lambda calculi, control operators, computer-aided developping of formal proofs, program extraction from proofs, computational content of classical logic.

A.13.1 The Novi Sad – Belgrade subsite

The Foundations of Computer Science Group (FCSG) is a joint research group of the Mathematical Institute of Serbian Academy of Science and Arts and University of Novi Sad.

The senior members are

- Silvia Ghilezan, contact person (Novi Sad)
- Kosta Došen (Belgrade)
- Mirjana Borisavljevi'c (Belgrade)
- Zoran Petri'c (Belgrade)

The FCSG has recognized expertise in type theory, categorial proof theory and associated application.

Regarding TYPES working group FSCSG has a long-term fruitful cooperation with the Turin group and several sites and subsites of the TYPES working group: Nijmegen, Lyon, Toulouse and Padova. Long and short-term visits to Nijmegen and Turin resulted in several published joint papers with Henk Barendregt and Mariangiola Dezani-Chiancaglini, which were communicated at various meetings including TYPES 2002, held in Nijmegen.

The groups recent focus is in foundational studies of computational aspects of classical logic and program extraction from classical proofs. The main goal is to employ intersection types in program analysis of classical proofs and to develop an intersection type assignment system which will completely control termination of classical programs.

A.14 The Udine site

The *Formal Methods and Logics of Computation* (FMLC) group at the Department of Mathematics and Computer Science of the University of Udine, has been involved in all four previous TYPES projects founded by the EU. The FMLC group has international reputation in type theory, lambda calculi, Logical Frameworks, Non Wellfounded Set Theory, Coinductive and Coalgebraic methods, and Exact Real Numbers computation. Recent work has focussed on the syntax and semantics of programming languages and the algebraic, categorical, logical, and geometric models which support the verification and analysis of programs, as well as the areas of Game Semantics and stepwise data refinement.

The group plans to contribute to the following objectives of this proposal.

Correctness of Computer Systems Computer arithmetic through our ongoing work on exact real computation using coinductive types. Programming language features: we have formalized in Coq an object-oriented language with side effects, and proved some important properties. We plan to continue this line of research, by developing certified tools (e.g., interpreters) for this kind of languages. Another line of research on this issue is the use of coinductive principles for reasoning on object oriented languages.

Foundational research Logical Frameworks: we are active in the area of Higher Order Abstract Syntax and binding, especially with regard to structural induction and recursion over abstract syntax. We are also working on foundational results unifying inductive and coinductive definitions, and covering many naturally occurring examples. We plan to investigate more liberal principles for such definitions.

The senior members of the group are

- Marino Miculan (contact person)
- Furio Honsell
- Pietro Di Gianantonio
- Marina Lenisa

Short CV for one key researcher: Furio Honsell. Laurea in Mathematics (1981, Pisa University), Diploma in Mathematics (1983, Scuola Normale Superiore, Pisa). Full professor in Computer Science at Udine University since 1990. Previous research and academic positions at Torino University (1983-1985), Edinburgh University (1986-1988). Rector of the University of Udine since 2001. Site Leader of Italian projects COFIN97, COFIN99 and EC HCM Lambda Calcul Typé, EC SCIENCE MASK, ESPRIT WG TYPES. Coordinator of the project COFIN01 "CoMeta". PC member and invited speaker to many international conferences. Member of the editorial board of MSCS, and of the IFIP WG 2.2. Research interests: semantics of programming languages, program logics, lambda-calculus, type theory, logical frameworks, logical, topological and categorical methods in informatics.

A.14.1 The Padova subsite

The logic research group in Padova has deep experience in using intuitionistic type theory for the development of mathematics, in particular formal topology. Its interest includes related matters such as normalization in type theories, connections with categorical universes, tools for user-friendly mathematical notation and proof search. This work has led also to basic logic, bringing a unified treatment to various constructive systems. The group has been active in TYPES since 1996, and organized the

Second Workshop on Formal Topology in April 2002, a TYPES small workshop. The group has on-going collaboration with the TYPES sites at Göteborg, München (LMU), Torino, Stockholm, Sophia Antipolis and Birmingham.

The group plans to contibute mainly to the Formal Mathematics objective of this proposal, with our work on formal topology, which we also plan to apply to computational mathematics and to theoretical computer science (e.g. models for concurrency). We also plan to organize the Third Workshop on Formal Topology in 2005 or 2006.

The senior members of the subsite are:

- Giovanni Sambin, contact person
- Silvio Valentini
- Claudia Faggian
- Maria Emilia Maietti

A.15 The Warsaw site

The Applied Logic Group is a part of the Institute of Informatics of Warsaw University. The group participated in the previous TYPES project *Computer-Assisted Reasoning based on Type Theory* in 1999–2003.

The expertise of the group as a whole is mostly in logics of programs, typed lambda calculi, type inference, finite model theory, theory of tree automata and mu calculus. Recently there is a growing interest towards proof assistants, especially the Coq system. This has already resulted in two PhD works prepared in collaboration with Paris Sud.

The group plans to contribut to the following objectives of this proposal.

Foundations work on inductive definitions, subtyping, termination, rewriting and modules.

Correctness of computer systems work on programming with dependent types.

Formal mathematics and education case studies.

We plan to continue and expand the collaboration with Paris Sud, INRIA and other sites, including possible further joint PhD projects.

The members of the group who will participate in the project include:

- Paweł Urzyczyn, contact person
- Jerzy Tiuryn
- Aleksy Schubert
- Daria Walukiewicz-Chrząszcz
- Jacek Chrząszcz
- Zdzisław Spławski (afilliated with Wrocław University of Technology)

Short CV for one key researcher: Pawel Urzyczyn. Born: March 31, 1954 in Warsaw, Poland. Education: MSc. (Warsaw University), 1978, Ph.D. (Warsaw University), 1983, Habilitation (Warsaw University), 1991. Title of professor, 2003. Position: Extraordinary Professor, since 1993. Area of research: theory of program schemes, logics of programs, typed lambda calculi, proof theory, type inference. Papers: journals-26, proceedings-11. Long stays abroad: Boston Univ. 1996; Kobenhavns Univ. 1997-8,2003-4. Member of Fundamenta Informaticae editorial board. PC member: TLCA'99, MFCS'99, FOSSACS'01, LiCS'01 and TLCA'03.

A.15.1 The Kraków subsite

The Foundations of Computer Science Group is a part of the Institute of Informatics of Jagiellonian University of Kraków.

The members of the group who will participate in the project include:

- Marek Zaionc, contact person
- Małgorzata Moczurad
- Paweł Waszkiewicz

Marek Zaionc and Małgorzata Moczurad work on lambda definability problems in typed lambda calculus. Marek Zaionc recently also works on asymptotic properties of propositional logics. Paweł Waszkiewicz works on quantitative aspects of domain theory and its implications for topology.

The group cooperates with the sites at ENS Lyon and Chambery. There are also good contacts with Birmingham University resulting in the PhD thesis of Paweł Waszkiewicz.

A.16 The IoC Tallinn site

The senior person involved is

- Tarmo Uustalu, contact person

He is part of the Software Department of the Institute of Cybernetics (IoC) in Tallinn and participated in the previous TYPES project through the Minho subsite of the INRIA Sophia site.

The group will mainly contribute to the project objective Foundational Research. Uustalo's current interests are programming with nested datatypes, representation and manipulation of non-wellfounded syntax and syntax with variable binding.

He is collaborating with R. Matthes, A. Abel (TU München), L. Pinto and colleagues (Minho), G. Barthe (INRIA Sophia). Uustalu and Matthes have collaborated intensively since 2001 (incl. several visits). The cooperation with Barthe started at Minho in 2000 and has by now become a project within the Estonian-French programme Parrot. In Nov. 2002, Uustalu visited M. Bezem (Bergen). In March 2003, Uustalu visited T. Altenkirch (Nottingham); he will visit IoC in Sept. 2003 to teach a course supported by the Estonian IT Foundation.

Short CV for one key researcher: Tarmo Uustalu was born 1969, MSc from Tallinn TU 1992, PhD from KTH Stockholm 1998, postdoc experience from U of Minho 2000-2002, currently senior researcher at IoC Tallinn. Main scientific interests broadly: proof theory, type theory, categorical logic, programming language semantics, functional programming. One of the main organizers of the international Estonian Winter Schools in Computer Science (EWSCS); has obtained a number of research/mobility grants from the Estonian Science Foundation, Estonian IT Foundation and foreign bodies; scientific person in charge for the FP5 IST accompanying measures project eVikings II, leader of a member node in FP5 IST thematic network APPSEM II. Participated in the previous TYPES project via the U of Minho subnode of the INRIA node.

References

[1] T. Altenkirch, W. Naraschewski, and B. Reus, editors. *International Workshop, TYPES'98, Kloster Irsee, Germany, March 27-31, 1998, Selected Papers*, volume 1657 of *LNCS*. Springer-Verlag, 1999.

[2] H. Barendregt and T. Nipkow, editors. *International Workshop TYPES'93, Nijmegen, The Netherlands, May 24-28, 1993. Selected Papers*, volume 806 of *LNCS*. Springer-Verlag, 1994.

[3] S. Berardi and M. Coppo, editors. *International Workshop, TYPES'95, Torino, Italy, June 5-8, 1995 Selected Papers*, volume 1158 of *LNCS*. Springer-Verlag, 1996.

[4] ESPRIT Basic Research Action 6453 - TYPES. `www.dcs.ed.ac.uk/lfcsinfo/research/types_bra/index.html`

[5] P. Callaghan, Z. Luo, J. McKinna, and R. Pollack, editors. *International Workshop, TYPES 2000, Durham, UK, December 8-12, 2000. Selected Papers*, volume 2277 of *LNCS*. Springer-Verlag, 2002.

[6] T. Coquand, P. Dybjer, B. Nordström, and J. Smith, editors. *International Workshop TYPES'99, Lökeberg, Sweden, June 12-16, 1999, Selected Papers*, volume 1956 of *LNCS*. Springer-Verlag, 2000.

[7] G. Dowek. *Voulez vous jouer avec les maths?* Le Pommier, 2002.

[8] P. Dybjer, B. Nordström, and J. Smith, editors. *International Workshop TYPES'94, Bastad, Sweden, June 6-10, 1994. Selected Papers*, volume 996 of *LNCS*. Springer-Verlag, 1995.

[9] H. Geuvers and F. Wiedijk, editors. *International Workshop, TYPES 2002, Berg en Dal, The Netherlands, April 24-28, 2002, Selected Papers*, volume 2646 of *LNCS*. Springer-Verlag, 2003.

[10] The Grammatical Framework.
`www.cs.chalmers.se/~aarne/GF/`

[11] E. Gimenez and C. Paulin-Mohring, editors. *International Workshop TYPES'96, Aussois, France, December 15-19, 1996 Selected Papers*, volume 1512 of *LNCS*. Springer-Verlag, 1998.

[12] G. Huet and G. Plotkin, editors. *Logical Frameworks: First International Workshop on Logical Frameworks, Antibes, May, 1990*. Cambridge Univ. Press, 1991.

[13] G. Huet and G. Plotkin, editors. *Logical Environments: Second International Workshop on Logical Frameworks, Edinburgh, May, 1991*. Cambridge Univ. Press, 1993.

[14] J-L Krivine. Intelligence & lambda calcul. *Science et Vie*, 1013, February 2002.

[15] TYPES Summer School'02, September 2002.
`www-sop.inria.fr/certilab/types-sum-school02/`

[16] Information on the TYPES proof systems.
`www.dur.ac.uk/TYPES/systems.html`

[17] Esprit working group 21900 - TYPES.
`www.cs.chalmers.se/ComputingScience/Research/Logic/TypesWG/`.

[18] Working Group 29001: Computer-Assisted Reasoning based on Type Theory (TYPES).
`www.dur.ac.uk/TYPES/`