Title page

INFORMATION SOCIETY TECHNOLOGIES
(IST)
PROGRAMME

Contract for:

Concerted Action/Thematic Network

Annex 1 - "Description of Work"

Project acronym: APPSEM-II
Project full title: IST Working Group: Applied Semantics
Proposal/Contract no.: IST-2001-38957
Related to other Contract no.:

Date of preparation of Annex 1: April 28, 2003

Operative commencement date of contract:

Contents

1 Project Summary

CA & TN CPF Form – Form A1

EUROPEAN COMMISSION
RESEARCH DIRECTORATES
GENERAL
CA & TN CPF FORMS

EN B 1 FP5CA/TN □□

FOR COMMISSION USE ONLY □□

Project Acronym [2]	APPSEM-II	**Proposal No** [3]	IST-2001-38957

A1. Project Summary [21]

Objectives (maximum 1000 characters)

Principal objective: Maintain and further develop an existing European network for research and technology transfer in the field of application-oriented semantics of programming languages.

Specific objective 1: description of existing programming language features (e.g. object-orientation, aspect-orientation, distribution, web-related)
Specific objective 2: design of new programming language features (e.g. programming with continuous data, resource-bounded computation, programming with dependent types)
Specific objective 3: transformation and generation of programs (e.g. partial evaluation, runtime code generation)
Specific objective 4: specification and verification of programs (e.g. logic for OOP, software model-checking)

Description of the work (maximum 2000 characters)

The objective of this proposal is to maintain and further develop an existing European network for research and technology transfer in the field of application-oriented semantics of programming languages.
Programming languages are (next to algorithms and software engineering) one of the essential ingredients for successful software development. They therefore have a huge economic impact: better programming languages and a better understanding of the existing ones will lead to higher productivity, reduced maintenance, and increased software reuse. Europe is a hotbed of programming language research with many internationally respected experts both in semantic theory of programming languages and in implementation.
The theoretical results and expertise that are available in the consortium will be used to solve problems relating to programming languages. At the same time and enabled by these practical problems the theoretical toolbox will be further developed. Mores specifically, the work programme of the working group is centred around the following nine themes:

- **Program structuring: object-oriented programming, modules (e.g. type-safe support for modules, OOP, AOP, components)**
- **Proof assistants, functional programming and dependent types (e.g. programming languages with dependent types)**
- **Program analysis, generation and configuration (e.g. run-time code generation)**
- **Specification and verification methods (e.g. software model checking, testing)**
- **Types and type inference in programming (e.g. types and effects)**
- **Games, sequentiality and abstract machines (e.g. game-theoretic semantics for nondeterminism)**
- **Semantic methods for distributed computing (e.g. language support for privacy)**
- **Resource models and web data (e.g. resource-bounded computation, reasoning about linked data)**
- **Continuous phenomena in Computer Science (e.g. computing with real numbers)**

Milestones and expected results (maximum 500 characters)

We expect the working group to solve practical problems arising from programming languages and to deepen the understanding of programming language semantics.
This will be achieved by way of research visits, workshops and summer schools. More specifically, we plan:

- **General workshops with all participants in the first half of each year.**
- **Several topical workshops as detailed in the proposal**
- **A WWW-page listing results – a summer school**

The overall objective of this working group is to maintain and further develop an existing European network for research and technology transfer in the field of application-oriented semantics of programming languages.

For a long time European research on programming languages and their semantics was divided into theoretical work initially spawned from practical questions but in the meantime independent of applications and practical work carried out on an ad-hoc basis and uninformed by recent theoretical results. It was the mission of the precursor to this working group—APPSEM-I—to change this situation by bringing theoreticians into close contact with practitioners and users.

This mission has been fulfilled beyond expectations: we have seen several important scientific breakthroughs that would not have been obtained without the interaction fostered by APPSEM (application of CPS to program analysis, application of linear types to memory management, use of dependent types in programming, to name a few) and also the emergence of a large, yet close, community encompassing both practical and foundational research on programming language semantics.

It is now time to further capitalise on these results. We want to consolidate and extend the existing but still fragile community of European research into application-oriented programming language semantics and increase visibility and exploration of the scientific results and the community as a whole.

This will be achieved by sponsoring inter-thematic and theme-specific workshops and funding short courses and summer schools. Moreover, we plan to install and maintain a *Semantics Portal* which will serve as entry point for anyone interested in programming language semantics. Apart from project acitivities and reports it will contain links to other researchers, event schedules, bibliography, tutorials, etc.

2 Project objectives

Programming languages are the basic material from which every software product is built. They therefore have a huge economic impact: better programming languages and a better understanding of the existing ones will lead to higher productivity, reduced maintenance, and increased software reuse. Europe is a hotbed of programming language research with many internationally respected experts both in semantic theory of programming languages and in implementation.

Semantics of programming languages refers to a formal mathematical description of the effect of a program or a program fragment written in some programming language.

Obvious applications for semantics of programming languages include the justification of compilation steps, in particular optimising ones, and the definition and justification of verification systems and program logics.

In order for such applications to be feasible the semantics must achieve a good degree of abstraction and lead to new insights. For instance, a “semantics” defined by a reference compiler does not lend itself towards formally verified optimisations or indeed any kind of verification that goes beyond testing.

While simple imperative programming and most features of functional programming languages are semantically well-understood, the subtler parts of imperative programming such as concurrency, pointer aliasing and encapsulation, as well as many features of object-oriented programming remain elusive despite considerable progress in the last decade. Moreover, new programming language constructs appear on the scene such as mobility, interoperability and scripting, aspect-orientation, document description, etc. for which application-oriented semantic accounts are urgently required.

2.1 Specific Objectives

The proposal is structured into nine interdisciplinary themes, which were identified as particularly promising for profitable interaction between semantic theory and practice. These themes relate to the following general areas:

- description of existing programming language features
- design of new programming language features
- implementation and analysis of programming languages

- transformation and generation of programs
- specification and verification of programs

The themes are

A. Program structuring: object-oriented programming, modules.

B. Proof assistants, functional programming, and dependent types.

C. Program analysis, generation, and configuration.

D. Specification and verification methods.

E. Types and type inference in programming.

F. Games, sequentiality, and abstract machines.

G. Semantic methods for distributed computing.

H. Resource models and web data.

I. Continuous phenomena in Computer Science.

A more detailed description of the themes and the scientific work to be carried out within them may be found below in Section 9.7.

The Working Group will promote collaboration by funding travel between sites and by organising and funding workshops. We will conclude the project with a summer school.

There are several industrial groups involved in the project. Apart from contributing to the research in some cases (e.g. Microsoft Research at Cambridge), their role will be to follow research developments, to give feedback on the usefulness of the solutions provided by the academic researchers, and to suggest new directions as well as concrete questions arising in practice.

The Working Group is intended to be a forum for interaction between programming language practitioners and theoreticians in semantics is general. Good communication is ensured by the existence of a common scientific basis in the well-established theory of programming language semantics and by the desire to exploit this theory in practice.

Each theme has one or two *theme coordinators* who will be responsible for ensuring and reporting progress within the specific themes. They summarise the activity within their theme in the progress and final reports.

3 Participant list

Partic. Role	Partic. No.	Participant name	Country	Date enter project	Date exit project
CO	1	LMU Muenchen	D	Start of project	End of project
MB	2	Aarhus	DK	Start of project	End of project
MB	3	Birmingham	UK	Start of project	End of project
MB	4	Cambridge	UK	Start of project	End of project
MB	5	Chalmers	S	Start of project	End of project
MB	6	Copenhagen	DK	Start of project	End of project
MB	7	Darmstadt	D	Start of project	End of project
MB	8	Edinburgh	UK	Start of project	End of project
MB	9	Freiburg	D	Start of project	End of project
MB	10	Genova	I	Start of project	End of project
MB	11	INRIA	F	Start of project	End of project
MB	12	EPFL Lausanne	CH	Start of project	End of project
MB	13	INRIA Loria	F	Start of project	End of project
MB	14	Minho	P	Start of project	End of project
MB	15	Nottingham	UK	Start of project	End of project
MB	16	Oxford	UK	Start of project	End of project
MB	17	Paris 7	F	Start of project	End of project
MB	18	Pisa	I	Start of project	End of project
MB	19	QMUL London	UK	Start of project	End of project
MB	20	Tallinn	EE	April 2003	End of project

4 Contribution to programme/key action objectives

By improving programming languages and providing infrastructure for successful software development the results of the proposed work will contribute to all aspects of a user-friendly information society.

Understanding semantics of new programming language features is essential for large scale distributed software development and for the development of supporting tools.

More specifically, the proposal addresses key action IV: *Essential Technologies and Infrastructure* and in particular action line IV.3: *Technologies and engineering for software, systems, and services*.

5 Membership

Membership of the working group is initially confined to the proposing sites as detailed in Appendix A. In exceptional cases the steering committee can decide on inclusion of new sites or subsites, provided this can be accommodated within the budget of the working group. In particular, we will make efforts during year 1 to include individual researchers or teams from newly associated states.

Cooperation of the partners is ensured by workshops and individual visits between sites as detailed in B6. The work programme is grouped into nine themes across sites. Theme coordinators as explained in Section 9.6 are responsible for cooperation on the respective theme.

6 Community added value and contribution to EU policies

Europe is a hotbed of programming language research with many internationally respected experts both in semantic theory of programming languages and in implementation. One should not forget that all-pervasive paradigms such as object-orientation originated in Europe.

The project exploits and strengthens the strong European theoretical expertise both in semantics of programming languages and in logic applied to computing. In particular, it will continue to exploit the advances made by the first APPSEM working group as well as the earlier ESPRIT Basic Research Action CLiCS (Categorical Logic in Computer Science). As before, there are close interactions with the EU Working Group TYPES.

Virtually all of the results obtained in applied semantics within Europe have a clear international dimension. They typically involve partners and expertise from different countries. The aims of the working group can thus not be achieved at a national level or through bilateral agreements.

7 Contribution to Community social objectives

Programming languages are the basic material from which every software product is built. They therefore have a huge economic impact: better programming languages and a better understanding of the existing ones will lead to higher productivity, reduced maintenance, and increased software reuse.

8 Economic development and scientific and technological prospects

The scientific results achieved during the funded period will be disseminated in the form of the deliverables described in Part B, in particular we maintain a visible web page describing all activities and listing publications and software products. The summer school at the end of the funded period will promote dissemination to junior researchers. All this will maintain and further extend the European strength in applied semantics and programming language theory.

9 Work plan

The work plan is structured into six work packages describing the activities of the working group, a list of deliverables allowing for monitoring and reviewing of progress and success and descriptions of the scientific work to be undertaken under each of the nine themes.

9.1. Work package list

Work package No	**Work package title**	**Lead contractor No**	**Person months**	**Start month**	**End month**	**Phase**	**Deliverable No**
WP1	Annual general workshops	18	N/A	3	18	N/A	D1,D2,D3, D4,D5,D6
WP2	Theme-specific workshops	2	N/A	1	36	N/A	D7, D8
WP3	Education	11	N/A	11	36	N/A	D9,D10, D11,D12
WP4	Individual visits	3	N/A	1	36	N/A	D13,D10, D12
WP5	Management and dissemination	1	N/A	1	36	N/A	D14 + compulsory deliverables
WP6	Industrial liaison	5	N/A	1	36	N/A	D15,D16, D17,D18, D19,D20

9.2 Work package descriptions

Work package description

Work package number:	WP1	Start date or starting event:	Kickoff workshop in month 3

Objectives

- Presentation of and learning about results obtained in the working group.
- Giving opportunity for communication between researchers working in different fields and especially between theorists and practitioners.
- Catalysing collaboration.

Description of work

This work package comprises three annual general workshops.

The first of these workshops will serve as kickoff meeting and will thus be held within the first three months.

The second one will be held in months 12-15.

The third one will be held in conjunction with a summer school.

Each workshop will be open to all participants including representatives from industrial partners and selected outside observers. Other researchers may attend the workshops and give talks if sufficient space is available.

Each workshop will last for three days and consist of 2-3 invited talks and 20-30 contributed talks within the various themes. In addition there will be a panel discussion with industrial representatives, a brainstorming session, a business meeting (open to members), and a meeting of the steering committee.

Workshop proceedings will be made available at the workshop and on our WWW site.

Deliverables

- Annual workshops (D1,D2,D3)
- Annual workshop proceedings (D4,D5,D6)

Milestones and expected results

Workshops in months 3 and 15 and 30.

Work package description

Work package number:	WP2	Start date or starting event:	Month 1

Objectives

- Bringing together researchers working on a particular topic
- Promoting new research fields within the APPSEM-II themes

Description of work

Throughout the course of the project we will hold theme-specific workshops bringing together researchers working on a particular topic within one or across several of the themes. We are planning to hold a minimum of nine workshops on the following preliminary list of topics. A larger number of workshops around 12-13 may be expected.

- Semantic foundations of aspect-oriented programming (A, E)
- Type systems for memory management (Themes C, H)
- Exact real number computation, domain theory, and formal topology (Theme I)
- Typing and program optimisation (Themes B, C, E, H)
- Security issues (Themes C, G)
- Nominal calculi and higher-order syntax (Theme B, E)
- Dependent types in programming (Theme B)
- Spatial logic (Theme H)

Further workshop topics may arise during the course of the project. These workshops will be spread evenly over the lifetime of the working group. The results presented at each workshop will be published on our web page or appear as formal proceedings in an appropriate scientific forum.

Deliverables

- Theme specific workshops (D7)
- Workshop proceedings (D8)

Milestones and expected results

A minimum of nine workshops throughout the project and two workshops per year will be held.

Work package description

Work package number:	WP3	Start date or starting event:	Month 1

Objectives

- Disseminating results obtained in the working group
- Training of young researchers

Description of work

We plan to hold a summer school open to postgraduate students and junior researchers at the end of the funding period to disseminate the results obtained in the working group. In addition to this we envisage short courses to be held by visiting researchers at their host site on their individual research topics.

Deliverables

- Organisation of summer school (D9)
- Organisation of short courses (D10)
- Lecture notes (D11)
- Course notes (D12)

Milestones and expected results

- Summer school during months 30–36.
- Short courses after month 15.

Work package description

Work package number:	WP4	Start date or starting event:	Month 1

Objectives

- Enabling collaboration between individual researchers on a particular topic.
- Creating possibility for short courses.

Description of work

We will also 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 summarised in the *annual progress reports*.
Such visits can be funded only if the visiting researcher hold a short course at the hosting institution, see D10.

Deliverables

- Publications and technical reports. (D13)
- Short courses. (D10)

Milestones and expected results

We expect a minimum of two such visit per year and a minimum of one research paper or technical report directly based on the results of such a visit.

Work package description

Work package number:	WP5	Start date or starting event:	Month 1

Objectives

- Project management
- Dissemination and evaluation

Description of work

The management structure is as detailed in Section 9.6. It comprises in particular regular meetings of the *steering committee* and business meetings open to all participants. Orthogonal to that *theme coordinators* are responsible for ensuring progress on their particular theme. They liaise with and respond to the steering committee. Dissemination of results is ensured via a *project web page* containing (links to) all other deliverables and results as well as other project-related documentation and serving as a “semantics portal” for the wider community.

Other means of dissemination are listed in WP3 (summer school and short courses) and WP6 (industrial liaison).

Deliverables

- WWW-site (D14)

Milestones and expected results

Work package description

Work package number:	WP6	Start date or starting event:	Month 1

Objectives

- Identify industrial challenges related to semantics
- Obtain an industrial perspective on APPSEM's research
- Disseminate information on applications of semantics in industrial practice

Description of work The industrial participants in APPSEM make up the Industrial Advisory Panel. At each annual general workshop, the panel will lead a discussion on Industrial Challenges to Semantics, which in our experience is an effective way to meet the first two objectives above. A summary of the discussions will be published on the APPSEM web site for later reference. To meet the third objective, industrial participants will be invited to speak on their work at the annual workshops.

Deliverables

- Reports on industrial challenges to semantics (D15, D16, D17)
- Presentations of invited industrial talks (MS PowerPoint or similar) to be published on our WWW site. (D18,D19, D20)

Milestones and expected results
Invited talks, panel discussion and reports in months 3 and 15 and 30.

9.3. Deliverables list

Deliverable No	Deliverable title	Delivery date	Nature	Dissemination level
D1	Annual general workshop	6	O	PU
D2	Annual general workshop	18	O	PU
D3	Annual general workshop	30	O	PU
D4	Proceedings of annual general workshop	6	R	PU
D5	Proceedings of annual general workshop	18	R	PU
D6	Proceedings of annual general workshop	33	R	PU
D7	Theme-specific workshops	36	O	PU
D8	Proceedings of theme-specific workshops	36	R	PU
D9	Summer school in year 3	36	O	PU
D10	Short courses	36	O	PU
D11	Lecture notes for summer school	36	O	PU
D12	Course notes for short courses.	36	O	PU
D13	Scientific publications	12	R	PU
D14	WWW-site	3	O	PU
D15	Report on industrial challenges	4	R	PU
D16	Report on industrial challenges	16	R	PU
D17	Report on industrial challenges	31	R	PU
D18	Presentations of industrial talks at workshop	4	O	PU
D19	Presentations of industrial talks at workshop	16	O	PU
D20	Presentations of industrial talks at workshop	31	O	PU

Description of deliverables

D1. Annual general workshop in month 3

D2. Annual general workshop in month 15.

D3. Annual general workshop in month 30.

D4. Proceedings of annual general workshop in month 3 to be publicised on the project WWW-site or as a special issue of an appropriate scientific journal.

D5. Proceedings of annual general workshop in month 15.

D6. Proceedings of annual general workshop in month 33.

D7. Theme specific workshops. A minimum of nine theme-specific workshops will held with a minimum of two per year as detailed in work package

D8. Theme specific workshop proceedings. For each theme-specific workshop proceedings will be made available prior to the workshop on our WWW-site and where appropriate polished proceedings will appear in scientific fora.

D9. Summer school in year three.

D10. Short courses held by visiting researchers.

D11. Lecture notes for summer school.

D12. Course notes for short courses.

D13. Scientific publications, technical reports. Scientific results obtained within the working group will be reported in technical reports or published papers and listed on our webpage.

D14. 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 “semantics portal” giving access to relevant online 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.

D15. Report on industrial challenges in semantics. A summary of the industrial panel discussion at the first annual general workshop.

D16. Report on industrial challenges in semantics. A summary of the industrial panel discussion at the second annual general workshop.

D17. Report on industrial challenges in semantics. A summary of the industrial panel discussion at the third annual general workshop.

D18. Presentations of invited industrial talks at first annual general workshop (MS PowerPoint or similar) to be published on our WWW site.

D19. Presentations of invited industrial talks at second annual general workshop.

D20. Presentations of invited industrial talks at third annual general workshop.

9.4 Project planning and timetable

Table 1

9.5 Graphical presentation of project components

Not applicable as components are largely independent.

9.6 Project management

Steering committee. The management structure we propose is a *steering committee* chaired be the coordinator and comprising one nominated senior scientist (*site leader*) from each site and one industrial representative.

The site leader will also be the scientific official in charge of the work and thus be responsible for administrative matters at the site including contacts with subsites.

participant no.	Site	Site leader
2	Aarhus	O. Danvy
3	Birmingham	A. Jung
4	Cambridge	G. Bierman
5	Chalmers	P. Dybjer
6	Copenhagen	N. Jones
7	Darmstadt	T. Streicher
8	Edinburgh	I. Stark
9	Freiburg	P. Thiemann
10	Genoa	G. Rosolini
11	INRIA	D. Rémy
12	Lausanne	M. Odersky
13	Loria	J.-Y. Marion
14	Minho	J. Saraiva
1	Munich	M. Hofmann
15	Nottingham	G. Hutton
16	Oxford	S. Abramsky
17	Paris	P.-L. Curien
18	Pisa	G. Ghelli
19	QMUL	P. O'Hearn
20	Tallinn	T. Uustalu

Each annual general meeting will comprise a *business meeting* of the steering committee where scientific progress will be assessed, necessary changes in direction will be proposed and adopted, and other issues of major importance will be discussed. The scientific part of the business meeting will be open to all participants.

Additionally, *extraordinary meetings* of the steering committee will be held in order to resolve possible conflicts arising during the project which require urgent attention. Unless required otherwise by the nature of the problem such meetings will be held in the form of email or telephone conference. Absent or unavailable members may be replaced by another member. Meetings of the steering committee will be convened by the coordinator. When no unanimous decision can be reached the coordinator will hold a majority vote. Decisions made at such meetings will be binding for all partners.

Subsites. Some teams include researchers from associated subsites with strong links to the researchers at the main site in question.

Site	Subsite	Scientist in charge at subsite
Aarhus	Aalborg	B. Thomsen
Birmingham	Sussex	B. Reus
Minho	Lisbon	J. Caires
Nottingham	Utrecht	J. Jeuring
Nottingham	Bonn	R. Hinze
Nottingham	Leicester	N. Ghani
Copenhagen	ITU	L. Birkedal
Copenhagen	DTU	H. Nielson
QMUL	Imperial	A. Edalat
QMUL	Bath	D. Pym

The budget for those main sites which have an associated subsite includes funding for the travel, meeting, and communication costs for the associated site, as well. These funds will be operated by the main site in question.

Industrial and associated sites The working group includes a number of industrial sites. The industrial site leaders form the *industrial advisory board* which sends one representative to the steering committee.

Apart from contributing to the research in some cases (e.g. Microsoft Research at Cambridge), the role of the industrial partners will be to follow research developments, to give feedback on the usefulness of the solutions provided by the academic researchers, and to suggest new directions as well as concrete questions arising in practice.

Site	Site leader
Microsoft	A. Gordon
Ericsson	T. Arts
Carlstedt	L. Augustsson
Prolog Development	T. Lindner

One academic site in Switzerland (EPF Lausanne) will not hold any funds but otherwise participate as ordinary site.

Coordination of research For each theme we have chosen a senior team member as *coordinator* who will be in charge of monitoring and reporting progress.

Theme	Coordinator
A	D. Rémy
B	T. Coquand
C	N. Jones
D	U. Reddy
E	F. Henglein
F	P.L. Curien
G	G. Winskel
H	P. O'Hearn, P. Gardner
I	A. Jung

Methods for ensuring good communication The following table summarises the involvement of individual sites with specific themes:

	A	B	C	D	E	F	G	H	I
Aarhus	X	X	X		X			X	
Birmingham	X			X	X				X
Cambridge	X	X			X	X		X	
Chalmers		X	X	X		X			X
Copenhagen		X	X		X			X	
Darmstadt	X								X
Edinburgh	X	X		X		X	X	X	X
Freiburg		X	X	X	X				
Genoa	X		X	X	X				
Lausanne					X				
Loria	X	X	X	X	X	X	X	X	X
Minho		X	X		X	X			
Munich	X	X		X				X	
INRIA	X	X	X	X	X			X	
Nottingham	X	X	X	X	X		X		
Oxford	X							X	
Paris						X			
Pisa			X	X	X		X	X	
QMUL	X	X	X	X	X	X	X	X	
Tallinn		X	X		X		X		

Theme coordinators will encourage and support communication between sites about issues regarding their theme.

World Wide Web The Working Group will have a web page with information about activities: meetings, publications, site descriptions, etc.

9.7 Description of themes

We describe the scientific objectives of each of the nine themes that structure the scientific work to be carried out. While the actual scientific work will be funded by academic resources or dedicated research grants the role of APPSEM-II is to

A. Program structuring: object-oriented programming, modules

Real-world software projects are hampered by the increasing complexity of the code involved, increasing both the development and maintenance costs. Software Engineers typically employ some form of program structuring as an attempt to tackle this problem. This is achieved by building large systems from other smaller code components, which are *parameterized*, either explicitly or implicitly, so that they can be *specialized* or *refined* while being reused. To ensure safety, every unit must specify precisely its accessible features and their properties through an *interface*. Then, *units* can be implemented and modified independently of one another while preserving their interfaces; they can also be safely assembled together by only checking their interfaces.

However, this general approach still lacks powerful mechanisms for both building separate units by combining several forms of parameterizations together (abstraction, subtyping, recursion, or object identity, which are only well-understood in isolation) and for assembling them together. Moreover, beyond well-typedness, which has been the focus of most previous works, reasoning principles and mechanical proofs and analyzes for object-oriented programs are still to be settled. Our aim is to tackle these shortcomings, and develop a better understanding of program structuring. We hope to bring together researchers in various areas (modules, objects, types, analyzes), ultimately aiming to transfer our results to real-world programming languages through our industrial partners. We have identified the following three main topics for research.

Bringing modules and objects closer will be tackled by studying, in the first case, several extensions of up-to-date modular and object-oriented systems: (1) polymorphic extensions of structures (2) Mixing modules (2) packagings, and (3) overloading. We then plan a comparative study of these resulting approaches, with the aim of developing a unified framework.

Mechanical proofs and analyzes for object-oriented programs will include the verification and inference of unit's interfaces. We shall also develop lightweight, core object calculi (containing e.g. assignments, block-structure, and object identity) into which we can compile higher-level object-oriented languages.

Reasoning principles for object-oriented languages will be developed. This includes adapting reasoning techniques, such as operational characterizations of behavioral equivalences, now well-understood for functional calculi, to our core object calculi. In particular, we shall explore semantic methods using recently-developed resource models. One important application is to use these principles to reason about compilation and linking.

B. Proof assistants, functional programming, and dependent types

The unifying theme of this research strand is the Curry-Howard analogy between proofs and programs, and its refinement due to Martin-Löf (Stockholm), Nordström, and Smith (Chalmers) revealing a close correspondence between key concepts in functional programming (constructors, evaluation to weak-head normal form, program, ...) and key notions in proof theory (proof in introduction form, reduction to introduction form, admissible rule, ...). This theme enables a number of applications to programming and computer-aided threorem proving.

The work to be carried out within APPSEM-II under this theme is structured in six topics:

- Termination of functional programs and of proofs.
- Subtyping
- Record types
- Constructor subtyping
- Typing of object-oriented languages
- Polytypic programming
- Normalisation of terms with variables
- Continuation-passing style
- Dependent types in programming

C. Program analysis, generation, and configuration

APPSEM-I already gathered a substantial expertise in program analysis and transformation. The following topics include areas in which previous accomplishments will be further advanced under APPSEM-II, plus several promising new directions that lie squarely within the area Program analysis, generation, and configuration.

- Aspect-oriented programming
- A calculus for meta-computations
- Compiling
- Continuations
- Efficient programs from proofs
- Normalisation by evaluation
- Partial evaluation and program generation

- Programming the temporal dimension of computation
- Resource boundedness and non-interference
- Taxonomy and forestation within the world of programs
- Termination analysis
- Type specialisation

D. Specification and verification methods

An important application of programming language semantics is the development and validation of reasoning principles for programs. A close study of semantics suggests novel methods for specifying the behaviour of software systems and components which in turn gives rise to new programming logics and program verification methods. The following topics will be further investigated under APPSEM-II

- Reasoning principles based on the logic of Bunched Implications (Pym, O'Hearn)
- Hoare-style logics with implicit representations of store
- Reasoning about information hiding and data refinement
- Specification and verification using dependent types
- Modelling languages and model-checking algorithms for probabilistic systems
- abstraction of liveness properties in the temporal logic CTL

E. Types and type inference in programming

Types in programming languages have, over the last 40 years, become such an integral part of programming, that it is hard to isolate them from other aspects of programming and programming languages.

Types are a conceptual and technical substrate in programming languages with multiple facets: (Static) type systems provide early detection, localization and prevention of errors, which nowadays is considered a *condition sine qua non* in complex software system construction. They provide enforced abstraction with explicit and well-defined interfaces that is key to disciplined and predictable composition and reuse of system components. They enable reasoning — both informally and through automated tools — about software at a higher level, using other models than bits and bytes. Also, because they not only capture how software is constructed but also how it may be used, they provide the substrate for reasoning in a modular fashion about systems in a manner where the intended or possible uses are taken into account.

Powerful type inference techniques have turned out to be an important technology that minimizes a programmer's burden for providing explicit type annotations in programs and provides useful, nontrivial feedback to programmers about the programs they or others have written. Type-based program analysis is also used increasingly in program analysis, since types are a natural vehicle for modular analysis of programs with higher-order features. The use of languages with solid type systems can increase — or rather provide — basic security guarantees on the Internet; e.g., the majority of reported security breaches on the Internet could have been avoided if strongly typed programming languages had been used.

Type-related research will go on in several APPSEM themes, as well as outside APPSEM, notably in collaboration with EU Working Group TYPES. In this theme we will focus on important aspects of core issues regarding type systems and type inference in programming:

- How to design new type systems.
- How to increase the scope of type systems.
- How to increase the (logical) expressiveness of type systems and extend type inference.

Specifically we propose to pursue the following activities.

- Semantics-based design of type systems
- Type and effect systems
- Parametricity and type inference

F. Games, sequentiality, and abstract machines

The semantic understanding of sequentiality has been considerably further developed during the period of the APPSEM-I project. The main tools for this purpose were various notions of games, corresponding abstract machines and also to some extent realizability models. During the APPSEM-II period, we intend to further develop this research both on a foundational level and towards applications to program verification.

We plan to further this research in the following directions

- Positional games played on graphs to obtain finer-grained analysis of strategies
- Game semantics and continuations; relationships to Girard's *ludics*
- Game-theoretic analysis of probabilistic computation
- Enlarging the scope of model-checking techniques via game semantics

G. Semantic methods for distributed computing

In this theme we will develop mathematical tools for the formalization and analysis of distributed computation. We plan to meet the challenge of the lack of a *global* mathematical setting for distributed computation through the development of a domain theory for concurrency which can handle the intricate structure (e.g. of event structures and name generation) and equivalences (e.g. bisimulation) of dsitributed computation. By moving to new semantic frameworks we can accommodate the richer structure of distributed computation, characterise its operations abstractly, and systematise operational semantics through the study of rule formats for structured operational semantics. Our aim will be the development and investigation of a process language with the following features: higher-order processes, static scoping, dynamic creation of names. The design of the language will be based on that of existing languages but crucially guided by a denotational model, the importance of which will be established by studying its relationship to operational and logical semantics.

The models we develop are bound to suggest new type disciplines for distributed computation. Types for distributed computation will almost certainly help in developing logics for distribution (cf. themes 4 and 5). (One type-based approach could proceed by extracting logics in the manner of logic for domains.)

The challenge of security in distributed computation has recently led to event-based techniques in the verification of security protocols. The event-based proof techniques are rather low-level however, so we will face the problems of how to support higher-level reasoning through appropriate equational theories, logics and typing judgements to ensure security properties. (In particular, symbolic model checking approaches including minimization algorithms for bisimilarity in the presence of name generation will be explored.) We will work to identify and prove the relations between analyses of security protocols based on cryptography as a “black box” and those based on true cryptography and probability, connecting to Topic 9.

H. Resource models and web data

This theme centres on techniques for managing resource both locally and across widely-distributed systems such as the Web. We focus on three contrasting but related sections: **spatial logics**, **implicit complexity**, and **resource access and security policies** all of which have emerged during APPSEM-I. We will further pursue this work in the following directions:

- Fundamental theory behind spatial logics
- Theory and applications of pointer logic

- Creating query languages and programming languages for the Web
- Developing type checking, theorem proving and model checking technologies that can be used in applications to both resource modelling and the Web
- Applications of spatial logic to distributed and mobile code
- Developing the resource-based approach to implicit computational complexity, and investigate its connection to the account of resource allocation and deallocation provided by spatial logics
- Static estimation of the consumption of specific resources such as heap space, stack size, cache use, system calls
- Static analysis of security policies
- Relationships with aspect-oriented programming

I. Continuous phenomena in Computer Science

The work in this topic is situated at the boundary between Computing and Physics. Traditionally, the former deals with discrete structures, while physical systems are described with real and complex numbers. The research will contribute to this interface by studying the problem of computing with exact real numbers and probabilistic system evaluation.

Regarding the former, we will build on substantial insight that was gained during the lifetime of APPSEM-I. This work has led to a number of semantic models which allow a closer analysis of the inherent

efficiency problems with exact real computation. In APPSEM-II, we will use this knowledge to define a refined model which allows for a controlled amount of intensional information being used during computations. We thus hope to be able to provide substantial help for groups which attempt to integrate exact computation into computer algebra packages. Links already exist with St Andrews to this end and we intend to use this connection to also reach their industrial partners specialising in computer algebra.

Within probabilistic system modelling and evaluation we propose to provide a semantic foundation for the work by Kwiatkowska and her group in Birmingham in probabilistic model checking. Their model checker PRISM is being used to evaluate industrial designs and this will give us an opportunity to reach out and transfer theoretical results to the market place.

10 Clustering

N/A

11 Other contractual conditions

Duration The project will last for three years.

Budget per participant We budget a total of €2700 (incl. 20% overheads) per individual participant to cover attendance of annual general meetings, specialised workshops, and individual visits. The decision between these possibilities is left to the participants and will be made on a case-by-case basis. One possible scenario is to attend two annual general meetings at a cost of €800 each, one specialised workshop colocated with another event at €250 and €400 to partially cover a one week visit to another site. Funds paid to members other than the coordinator are exclusively for travel, subsistence, and participation fees.

We budget for our industrial partners on the basis of two persons per partner and similar scenarios.

Number of participants per site The amounts requested for members are calculated according to their number of participants with 7 being the maximum for any particular institution.

Advance payment in month 1 We plan an initial advance payment in month 1 to members corresponding to €900 per person. Allocation of funds in years 2 and 3 will be based on the level of actual activity as well as possible growth or shrinkage of the consortium or particular sites.

No advance payment will be made to industrial partners as these are managed as subsites of the coordinator who will reimburse actual expenditures.

Distribution of funds after month 12 The distribution of funds for years 2 and 3 will be decided by the steering committee in months 11 and 17 for the following year. The committee may also decide to keep a certain portion of the funds with the coordinator to fund activities of participants on an individual basis.

Breakdown of costs A detailed breakdown of yearly costs per site may be found in the table on page 26. Please note that the figures for years 2 and 3 are subject to change according to decisions on re-allocation to be made by the steering committee. Also note that the overhead charges of 20% are estimated at this point. Any change in these overhead charges will not affect the amount of the initial payment in month 1.

Local expenses incurred by workshop organisers will be paid for through participation fees that are included in the above individual amount. We envisage a €250 fee for annual general meetings and €100 for individual workshops. These cover location, registration staff, and light lunches.

We plan to invite 3 speakers to every annual general meeting at a cost of €1300 each resulting in a total amount of €12000. This covers airfare at €425, 5 days hotel and subsistence at €900.

We budget another €12000 for the final summer school. This includes 4 invited lecturers at €6000 (airfare+accommodation at summer school site) and 10 student grants at €600.

We request €15000 plus overheads for coordination costs at LMU Munich. This includes €4816 to employ a part-time worker for 8hrs/week. His/her tasks will include internet presence, accounting, and correspondence. The remainder is composed of €4175 for a desktop computer and printer, €2500 for web-, mail- and fileserver provision, €1780 for consumables.

We stress that all personnel cost is for administration and coordination; no research is being paid for directly.

Breakdown of yearly costs

Table 2

A Consortium description

The consortium comprises a large majority of the leading researchers in the field within Europe both in academia and in industry. During APPSEM-I and prior EU-funded activities such as CLICS there exist a closely knit network of mutual relations manifesting themselves in exchange of postgraduate students, joint research papers, joint projects, both national and EU-funded.

The partners fulfill equal roles within the working group within the themes of their expertise. The working group's purpose is to continue exchange of ideas between researchers working on different but related fields, in particular practitioners and theoreticians, as well as to enable international collaboration on particular specialised topics.

The success of the APPSEM-I working group has amply demonstrated the catalytic effectiveness of the proposed structure.

Efforts will be made, especially during the first year, to include new members from associated states. There exist presently contacts with Tarmo Uustalu from Cybernetic Institute at Tallinn, Estonia.

A.1 The coordinator

The project is coordinated by Martin Hofmann, Ludwig-Maximilians-Universität (LMU), München. His main research interests include principles of programming languages, especially complexity-theoretic aspects; type theory and mathematical logic; formal methods. He has been a reader in Informatics at the University of Edinburgh from 1998-2001 and is a full professor for Theoretical Informatics at LMU since September 2001. He has participated in the APPSEM and TYPES working groups and is currently site leader for a EU-funded project (MRG) under the FET "global computation" initiative. He has organised several conferences and workshops, e.g., CTCS'99 in Edinburgh; he has served on a number of programme committees of international conferences such as LICS, FOSSACS, TLCA, TACS.

A.2 The partners

Each participating institution is described together with names of senior researchers and their themes of expertise as described in Part B. For convenience we relist the themes here

A Program structuring: object-oriented programming, modules.

B Proof assistants, functional programming, and dependent types.

C Program analysis, generation, and configuration.

D Specification and verification methods.

E Types and type inference in programming.

F Games, sequentiality, and abstract machines.

G Semantic methods for distributed computing.

H Resource models and web data.

I Continuous phenomena in Computer Science.

Aarhus The involved researchers at Aarhus include those permanently employed at the department of computer science together with appropriate expertise at BRICS. The senior researchers involved are:

- Olivier Danvy (site leader) (C, H)
- Ulrich Kohlenbach (B, C, H)
- Peter Mosses (C)
- Mogens Nielsen (D, G)
- Michael Schwartzbach (A, E)

The researchers in Subsite **Aalborg** include a number of permanent staff at the Department of Computer Science. The senior researches involved are:

- Bent Thomsen, site leader
- Lone Leth Thomsen

Lone and Bent were involved in APPSEM as industrial partners while working for ICL in the UK. They joined the department of computing, Aalborg University, in April 2002. They both have a background in formal foundations, concurrency theory, programming languages, distributed and mobile systems, agents and industrial applications. Their current research interest are investigating what comes after established technologies such as C/C++, VB, C#, Java, CORBA, XML, SOAP, the web and even emerging mobile technologies like WAP. They believe that the next leap in computing will have strong formal foundations, based on integration of programming language concepts, run-time systems, distributed computing, security, mobility and web/WAP. Thus the work by the participants in APPSEM-II is of paramount importance to understanding and establishing the foundations for the future mobile and global communication and computation platform.

Aalborg is a subsite of Aarhus with strong collaboration ties, e.g. through BRICS.

BRICS (Basic Research in Computer Science) is funded by the Danish Research Foundation to investigate mathematical foundations of Computer Science, notably Algorithmics and Mathematical Logic, alongside existing activities in Semantics of Computation. BRICS is also a PhD School.

The involved researchers have considerable expertise in the area of semantics and implementation of programming languages, domain theory, concurrency, types, logic, proof theory, and web programming. The group has, and can draw on, additional expertise in category theory, algorithmics, and quantum computing.

Aarhus expects to contribute to themes A, B, C, E, and H.

BRICS maintains natural links with IKU/Copenhagen (Neil Jones, Andrzej Filinski, and Julia Lawall: partial evaluation, normalization by evaluation/type-directed partial evaluation, continuations), Chalmers/Goteborg (Peter Dybjer, Thierry Coquand: normalization by evaluation), Muenchen (Helmut Schwichtenberg, normalization by evaluation, program synthesis), and Genoa (Eugenio Moggi, program generation).

Also, this spring, one of Ian Stark's students at Edinburgh is visiting Olivier Danvy at BRICS under a Marie Curie Fellowship.

Ulrich Kohlenbach has natural links with Munich (Martin Hofmann, Helmut Schwichtenberg: implicit computational complexity, logic, proof theory).

Birmingham The research team at Birmingham includes members from the Theory of Computation Group as well as the Functional Programming Group. Senior team members are

- Achim Jung, site leader (I)
- Martín Escardó (I)
- Marta Kwiatkowska (D, I)
- Uday Reddy (A, E, F)
- Eike Ritter (A, F)
- Mark Ryan (D)
- Hayo Thielecke (E).

The University of **Sussex** at Brighton will form a subsite with principal researchers:

- Bernhard Reus, site leader (A, D)
- Guy McCusker (F)
- Matthew Hennessy (G)
- Jim Laird (F)
- Julian Rathke (A, G, H)
- Vladimiro Sassone (E, G).

The researchers at Sussex comprise the "Foundations of Computation" research group at the School of Cognitive and Computing Sciences. The common objective of this group is the development of semantic theories and description languages for object-oriented, concurrent, and distributed systems. Topics investigated in Sussex include abstract foundational calculi for mobile systems in which the concepts of location, mobility, failure and security play an central role (Hennessy,Rathke, Sassone), semantic approaches to control of interference in higher-order imperative languages (Laird, McCusker), and programming logics for object oriented languages (Reus).

One strand of research is to build on previous work with Ghica, and work of Ong, which provides an algorithmic understanding of game semantics, to produce program analysis methods for higher-order imperative programs. We envisage collaboration with the group at Oxford (Abramsky, Ong, Murawksi). A second strand takes inspiration from Curien and Herbelin's study of semantics-inspired abstract machines for the lambda-calculus. Our first aim is to develop such an abstract machine for an imperative language, based on the games models. A second goal is to understand more deeply the laziness inherent in sequential algorithms models, especially in the light of Laird's recent work on sequentiality (unpublished works: characterizing sequentiality in terms of stability, and understanding sequential algorithms via games). This could lead to a semantics-based fully-lazy implementation of the lambda-calculus.

Cambridge The research team at Cambridge includes both members of the Computer Laboratory and of the Department of Pure Mathematics and Mathematical Statistics.

The senior team members are:

- Gavin Bierman, site leader (A, B, E, H)
- Marcelo Fiore (G)
- Martin Hyland (F)
- Robin Milner (B, G)
- Alan Mycroft (E, H)
- Andrew Pitts (A, B, E)
- Peter Sewell (D, E, G, H)
- Glynn Winskel (G)

There are also a number of PhD students and postdoctoral researchers. The group expects to contribute to themes A, B, D, E, F, G and H.

The Theory and Semantics Group at Cambridge has an established international reputation for its work in theoretical areas of Computer Science, including considerable expertise in the area of programming language semantics. It has strong links with those involved with work in logic and category theory in the Department of Pure Mathematics, and also other more applied research groups in the Computer Laboratory, including the Systems Research Group and the Programming Research roup. It also has strong ties with the relevant members of the Microsoft Research Laboratory in Cambridge.

Chalmers

- Peter Dybjer, site leader (C, D, E, I)
- Marcin Benke (E)
- Koen Claessen (D)
- Catarina Coquand (E)
- Thierry Coquand (D, I)
- Jörgen Gustavsson (D)
- Rogardt Heldal (G)
- John Hughes (C, D, E)
- Patrik Jansson (E)
- Bengt Nordström (E)
- Dave Sands (G)
- Jan Smith (E, I)

The Chalmers APPSEM team will contribute to the themes A, D, E and I. The APPSEM team at Chalmers consists of members of three research groups:
The Functional Programming Group, the Programming Logic Group, and the ProSec Group (Programming-language based Security). The senior team members are: Marcin Benke, Koen Claessen, Catarina Coquand, Thierry Coquand, Peter Dybjer (site leader), Rogardt Heldal, Patrik Jansson, Jörgen Gustavsson, John Hughes, Bengt Nordström, Dave Sands, and Jan Smith.

The Functional Programming Group has had a leading role in the design and implementation of functional programming languages for many years. Several group members participated in the design of the standard lazy functional programming language Haskell and the Haskell compiler hbc. The group has also developed the random test case generation tool QuickCheck. Other areas of interest include type systems, partial evaluation and program transformation, abstract interpretation and type-based program analysis, improvement theory, graphical user interfaces, and formal hardware design.

The Programming Logic Group has long-standing experience and expertise in the area of Intuitionistic Type Theory and its application to programming and mechanization of proofs. The proof assistants ALF and AGDA, with the user interface ALFA was developed by this group. Current interests include record types for proof assistants, termination checking for proof assistants, automatic theorem proving, combining testing and interactive program verification, and formal topology.

The ProSec group is working on the solution of computer security problems using programming language based methods, in particular semantics, static analysis and program transformation. Recent work has focussed on the modelling and analysis of confidentiality properties of programs, and the elimination of covert channels in code.

The Chalmers APPSEM team will contribute to the following themes: A, E, D and I.

Copenhagen Groups doing APPSEM-related research at Copenhagen are in three locations:

- DIKU (University of Copenhagen);
- IT-C (IT-University of Copenhagen, not part of the University of Copenhagen); and
- Denmark's Technical University, DTU.

All the researchers at Copenhagen were involved, in one way or another, in Danish breakthroughs in the 1980s and 1990s in the field of program generation by partial evaluation. Their research activities have broadened since then to new but related areas, some more theoretical and some more practical, and focussing on topics such as types in programming language implementation and theory (Henglein, Filinski, Birkedal, Nielson, Nielson, Jones, Mogensen); applications of partial evaluation (Lawall); abstract interpretation (Nielson, Nielson, Jones, Mogensen); termination analysis (Jones, Nielson, Nielson); domain theory (Birkedal); foundations of computation (Grue, Jones); and complexity in relation to programming languages (Jones).

Senior researchers and interest areas include:

- Neil Jones, site leader (B, C, E, H)
- Andrzej Filinski (C, E)
- Klaus Grue (B)
- Fritz Henglein (C, E)
- Julia Lawall (A, C)
- Torben Mogensen (C, E)

Subsites: **IT University of Copenhagen** with key researcher

- Lars Birkedal (E, H)

Denmark's Technical University with key researchers

- Flemming Nielson (C, E)
- Hanne Nielson (C, E)

Darmstadt Key researchers

- Thomas Streicher, site leader (A, F, I)
- Klaus Keimel (I)

as well as two young temporary researchers

- Peter Lietz (I)
- Tobias Löw (F)

On the one hand classical domain theory and its application to computation on classical spaces is a traditional reserch subject represented by K. Keimel who is cooperating with A. Jung and M. Escardó from Birmingham.

On the other hand T. Streicher has been working in the fields of semantics of constructive logics (in particular type theories), realizability models (and its applications to sequentiality), exact real number computation and more recently on semantics and logic of object calculi. Streicher is cooperating with M. Escardó from Birmingham, J. Longley and A. Simpson from Edinburgh and B. Reus from Sussex.

Peter Lietz is working in the field of realizability models for constructive analysis and T. Löw is interested in sequentiality and universal models for linear and intuitionistic λ–calculi.

Edinburgh The research team at Edinburgh are associates of the Laboratory for Foundations of Computer Science, LFCS. Senior team members are

- Ian Stark, site leader (A, G, H)
- David Aspinall (B, G, H)
- Stephen Gilmore (H)
- John Longley (F, I)
- Gordon Plotkin (G)
- John Power (H)
- Don Sannella (D, H)
- Alex Simpson (I)
- Mark Steedman (H)
- Daniele Turi (G)

There are also a number of PhD students and junior postdoctoral researchers. The group expect to contribute to themes A, B, D, F, G, H, and I.

LFCS is an established research institute, which over the last 15 years has built a significant international reputation for fundamental investigations in theoretical computer science. Relevant expertise within the laboratory includes type theory, concurrency, semantics, formal specification, and the applications of category theory to computer science.

LFCS has strong European links, both current and historic, many relevant to the area of applied semantics. In particular, it is a member of the TYPES working group, and a site for the EU project *Mobile Resource Guarantees*.

The laboratory is a part of the Division of Informatics at the University of Edinburgh. This is the highest rated (5*A) and largest department for computer science research in the UK.

Freiburg Key researchers

- Peter Thiemann, site leader (E)
- David Basin (B, D)

The Freiburg group consists of members of two research groups, Programming Languages (Thiemann) and Software Engineering (Basin). Senior team members include Jan Smaus, Luca Viganò, and Burkhard Wolff.

Research in the programming languages group focuses on program transformation, partial evaluation, and type systems. A central research direction is on programmable type systems, in particular geared towards applications in domain specific languages. Research in the software engineering group focuses on formal methods for specifying, verifying, and more generally developing safe and secure systems. This includes both foundational work in specification languages, theorem proving, and development methods, and research on applications to particular languages and paradigms.

Altogether, the Freiburg site would contribute to themes B, C, D and E.

Genoa The key researchers in Genoa are all associated with DISI, and they are:

- Giuseppe Rosolini, site leader (D)
- Francesca Levi (C)
- Eugenio Moggi (C)
- Elena Zucca (A, C, E).

There are also PhD students and a number of junior researchers. The group will contribute at least to themes A, C, D, and E.

Research activities at DISI cover a selection of current areas in Computer Science. The most up-to-date technologies and their applications are investigated in a variety of research activities. Projects range from student-based exploratory topics to works funded by the main public research institutions (such as the Ministry of Research, the National Research Council, the Italian Space Agency, and the National Institute for the Physics of Matter), to large-scale cooperative research activities funded by the European Community. A significant amount of research efforts is devoted to applied research activities in cooperation with the industrial world.

The researchers involved in the project come all from the Programming Language group at DISI, whose current research topics include design and foundations of modular and object-oriented languages and systems, logic and category theory, Synthetic domain theory, foundations and calculi for meta-programming and staging, monadic metalanguages and computational effects.

The group has developed strong links with many departments at other universities, including Cambridge, Edinburgh, ITU Copenhagen, Pisa, QMUL London.

INRIA The participation of INRIA comprises the Cristal group at Rocquencourt, the group around Gilles Barthe at Sophia Antipolis, and IRISA at Rennes. The senior participants would be

- Didier Rémy, site leader (A,E, H)
- Gilles Barthe (B)
- Yves Bertot (B)
- Joëlle Despeyroux (B)
- Pascal Fradet (A, C, E)
- Marieke Huisman (A, D)
- Thomas Jensen (A, C, E)
- Xavier Leroy (A, B, E, H)
- Michel Mauny (E, H)
- François Pottier (A, E)
- Pierre Weis (E)

The Cristal group is developing the Caml programming environment, which includes the Ocaml language, but also advanced libraries and programming tools for the Ocaml language. A few examples of recent developments are CamlIDL (a stub code generator and COM binding), Bigarray (a module to manipulate large or disk-resident arrays) and Ocamldbm (an interface to databases). Some development around XML is also being started. The Cristal group has expertise in language design, compilation, module systems, types and type inference, program analysis, and security,

The Lande group has expertise in program analysis, verification and transformation of programs, types and type inference. One of their focus is the application of these techniques to program structuring, and in particular, to aspect-oriented programming. They also intend to use program analysis techniques to the verification and inference of security interfaces for program modules.

The Lemme-Miro group has expertise in type and proof systems, program verification and security. Their work particularly focuses on programming with dependent types within the Coq proof assistant (recent examples of large development include a compiler for a simple imperative language amd the JavaCard platform) and on the integration of leightweight proof techniques in programming languages (for example the use of extended static checking techniques in Java). The team is also active in designing novel type systems for programming languages, in particular for integrating higher-order abstract syntax in dependent type theory.

Altogether, the INRIA site would contribute to themes A, B, C, D, E, and H.

Lausanne Key researchers

- Martin Odersky, site leader (E)

Note by coordinator: A detailed description of the Lausanne site will be delivered later.

Loria Loria is a CNRS and INRIA computer science laboratory located at Nancy which includes three universities with a PhD School. The involved participants are members of the team Calligramme, Miró, Protheo and Types. The senior team members are

- Jean-Yves Marion, site leader (H)
- Olivier Bournez (D, I)
- Didier Galmiche (A, D, G, H)
- Isabelle Gnaedig (D, H)
- Philippe de Groote (B)
- Claude Kirchner (B, D)
- Hélène Kirchner (B, D)
- Fanois Lamarche (F, I)
- Luigi Liquori (A, B)
- Pierre-Etienne Moreau (C, D)
- Christophe Ringeissen (C, D)

The involved researchers of the four Loria teams have expertise in area of specification and verification, functional programming, and in particular in object oriented programming, compilation and program semantics, and complexity.

The object of Calligramme team (P. de Groote, J-Y Marion, F. Lamarche) is the development of tools and methods stemming from proof theory, and especially linear logic. In software engineering, the main subject of interest are categories, functional programming and implicit computational complexity.

One of the goals of Miró team (L. Liquori) consists in investigating the possibility to “reconcile” object-oriented programming and functional programming while keeping the spirit of the former and the elegance of the latter. We study type theory, new systems improving formal proofs, efficient compilation and execution of object-oriented languages, and object-oriented operating systems. We are interested in certifying the tools developed for these languages (interpreters, compilers, . . .), using the Coq system as a favourite proof assistant.

The Protheo team (O. Bournez, I. Gnaedig, C. Kirchner, H. Kirchner, P.-E. Moreau, C. Ringeissen) has been strongly involved in the specification and verification methods since the last ten years. More recently and in particular in the context of compilation technique for rule based languages, we also focus our attention on program analysis. Recently we become also active in the extensions of type theory to support the rewriting calculus for computation and deduction purposes.

The Types team (D. Galmiche) has expertise in constructive and substructural logics, from different points of view : proof-theory, proof-search, semantics, systems specification and verification. In the context of this project, we will focus on the study of resource logics and the design of new theorem proving methods that provide proofs or countermodels in case of non-provability.

Minho The research team at Minho includes members from the Department of Computer Science as well as the Department of Mathematics. The senior team members are:

- João Saraiva, site leader (B, C)
- José Bernardo Barros (B , C)
- Luís Soares Barbosa (B , D)
- Maria João Frade
- José Nuno Oliveira (B , C)
- Jorge Sousa Pinto (B, F)
- Luís Pinto (E , F)

The involved researchers expect to contribute to themes B, C, E, and F.

Minho maintains natural links with Oxford (Abramsky: computational interpretation of sequent calculus - one of Abramsky PhD student is José Espirito Santo, a member of Minho Department of Mathematics. Oege: embedded domain specific languages and attribute grammars), INRIA (Barthe: constructor subtyping systems - one of Barthe PhD student is Maria João Frade, a member of our Department of Computer Science), and Nottingham (Backhouse: Oliveira is working on the application of the relational theory of datatypes to finite mapping-based data structures subject to complex invariants.)

Luís Pinto, Jorge Sousa Pinto, and José Espírito Santo have expertise in structural proof theory (e.g. Herbelin's sequent calculus) and in intensional approaches to the λ-calculus (e.g. geometry of interaction). Motivated by Curien and Herbelin's claims on"the relevance of sequent-calculus to the computational study of the λ-calculus", they are interested in pursuing the study of λ-calculi for the sequent calculus, particularly as a way of modelling aspects of the implementation of functional languages (like sharing and evaluation strategies), and as a basis for deriving abstract machines for the λ-calculus.

Barros, Oliveira and Saraiva intend to work on the analysis and transformation of functional programs for program optimization (deforestation techniques), program understanding (forestation techniques) and incremental evaluation, using both formal methods and attribute grammar-based techniques. Oliveira will pursue his work on data-refinement by generic transformation and on reverse transformations intended for program understanding and reverse-engineering.

Barbosa current research is oriented towards the use of coalgebraic methods in (reactive) program construction. In this context, Minho intends to contribute to the development of semantical models and calculi for coordination patterns and middleware.

Subsite: **Lisbon**

- Luis Caires (H)

Munich Key researchers are

- Martin Hofmann, coordinator (A, H)
- Andreas Abel [B)
- Jan Johannsen (H)
- Ralph Matthes (E)
- Helmut Schwichtenberg (B, H)
- Martin Wirsing (D)

The research team at Munich includes members from both the mathematical institute and the computer science institute. The Munich group will contribute to themes A, D and H. Hofmann, Johannsen, Schwichtenberg intend to work on programming language characterisations of parallel complexity classes such as AC, NC using methods from implicit computational complexity theory, in particular building on earlier results from our group and from Marion (Loria).

Hofmann will further pursue his work on static memory management for functional and object-oriented languages. This will be done in close cooperation with researchers from Edinburgh (Aspinall), QMUL (O'Hearn), Loria (Marion).

Abel and Matthes will cooperate with Göteborg researchers an the topic of termination checking (Theme B).

Building on earlier work by S. Merz (Munich) Hofmann and Wirsing will commence work on abstraction of liveness properties in the temporal logic CTL built into the theorem prover PVS. We will also investigate possible uses of PVS for object-oriented software verification with model checking thus pursuing earlier work by Hofmann and Tang.

Nottingham The research team at Nottingham includes members of the Foundations of Programming research group. It also includes Bonn, Leicester, and Utrecht as subsites. The members of the team have expertise in functional programming, generic programming, program verification and construction, type theory, and applications of logic and category theory.

Senior team members in Nottingham:

- Graham Hutton, site leader (B, D)
- Thorsten Altenkirch (A, B, D, E)
- Roland Backhouse (A, B, C, D)
- Louise Dennis (B, D)

Senior team members in **Bonn**:

- Ralf Hinze (B, E)

Senior team members in **Leicester**:

- Simon Ambler (B, D)
- Roy Crole (B, D)
- Neil Ghani (B, G)

Senior team members in **Utrecht**:

- Johan Jeuring (B,C,E)

Oxford Three research teams at Oxford are involved, representing a broad span of interests, ranging from abstract semantical theories to their application in concrete software artifacts:

- The Foundations Group
- The Algebra of Programming Group
- The Programming Tools Group

The senior researchers are

- Samson Abramsky, site leader (F, H)
- Alexandru Baltag (A)
- Richard Bird (A)
- Jeremy Gibbons (A)
- Oege de Moor (A)
- Andrzej Murawski (F)
- Luke Ong (F, H)

Postdocs that will participate in APPSEM-II include Corina Cirstea (A), Bob Coecke, Andrew Ker, Keye Martin (I), and Ganesh Sittampalam.

The planned contributions are arranged by theme:

- Program structuring: object-oriented programming, modules: Combined algebraic and coalgebraic approaches to object-oriented programming. (Cirstea) Coalgebraic logic, logics for multiple agents and complex information updates, and connections with security. (Baltag)
- Proof assistants, functional programming, and dependent types: The implementation of annotation-assisted type checkers for programming languages with dependent types. (De Moor)

  Dependent type checkers often fail to spot equalities between type expressions that are obvious to the applications programmer. It is therefore desirable that the programmer can give some hints to the type-checking algorithm, in the form of equations that the type-checker may use. In collaboration with Nottingham and Chalmers, Oxford will undertake the implementation of a variant of Haskell for experimentation with such features, and we shall study its semantics. Furthermore, we plan to investigate the application of such a language for designing embedded domain-specific languages.
- Program analysis, generation, and configuration: Formal proofs of program transformations (De Moor). This continues ongoing joint work with Copenhagen on an executable specification language for program transformations, where side conditions of transformations are formulated in temporal logic.

  Implementation techniques for staged programming languages, and the use of staging in numerical computing. (De Moor) This work will be carried out in collaboration with Microsoft, Cambridge, and Genoa.
- Games, sequentiality, and abstract machines: Algorithmic game semantics: we are looking at algorithmic formulations of game semantics as a basis for software model-checking and program analysis. (Abramsky, Ong, Murawski)
- Continuous phenomena in Computer Science: Abramsly, Coecke work on connections with quantum computing and measurement theory.
- Resource models and web data: linear logical and game-based approaches will be further developed by Murawski, Ong, Abramsky.

Paris

- Pierre-Louis Curien, site leader (F)
- Vincent Danos (F)
- Olivier Laurent (F)
- Paul-André Melliès (F).

Note by coordinator: A detailed description of the Paris site will be delivered later.

Pisa The research team at Pisa includes members of the Database, Object Technology, Program Analysis and Models and Languages for Open Distributed Systems groups. These groups have a strong research records in the APPSEM fields of object-oriented programming, functional programming, program analysis, specification and verification methods, type systems, semantic methods for distributed computing. The senior team members are:

- Giorgio Ghelli, site leader (H)
- Giuseppe Attardi (C, H)
- Pierpaolo Degano (C, G)
- Giorgio Levi (C, E)
- Ugo Montanari (D, G)

The group has strong links with the groups in Cambridge, Genoa, INRIA-Rennes and Copenhagen.

QMUL Queen Mary, University of London.

The research group at Queen Mary has strengths in logic, program analysis and concurrency. In particular, the logic of bunched implications, which can be seen to underlie the ”spatial logics” part of the Resource Models and Web Data theme, was developed at Queen Mary during APPSEM-I, and research on reasoning about pointers using the logic is going forward at a rapid pace. Other lines going forward include work on types for concurrency, categorical semantics, proof theory, and security.

The senior members at QMUL are

- Peter O’Hearn, site leader (A, C, D, H)
- Gianluigi Bellin (B, F)
- Kohei Honda (E, G)
- Pasquale Malacaria (E, F, G)
- Edmund Robinson (D, E, H)

Subsite **Bath**:

- David Pym (D, E, H)
- Carsten Führmann (D, E, H)

Subsite **Imperial college**:

- Abbas Edalat (I)
- Philippa Gardner (H)

Tallinn, IoC The senior researchers of the IoC Tallinn team are

- Tarmo Uustalu, site leader (B, E)
- Peeter Laud (C, G)
- Varmo Vene (B, C)

The research areas of the group are type theory and type systems, categorical logic, functional programming, and semantics-based program analysis. Uustalu, partly jointly with Vene, has worked on inductive and coinductive types in type theory and programming, type-based termination, generic programming, coalgebras and monads in programming language semantics, CPS and monad transforms, representation and manipulation of non-wellfounded syntax, syntax with variable binding. Vene and Laud have worked on various program analyses (such as points-to analysis in object-orientation and data race analysis in multithreaded computation). Laud has particularly been specializing in computationally secure information flow, for which work he recently won the EAPLS best paper award for papers presented at ETAPS 2003.

The team has strong links to LMU Muenchen (Ralph Matthes), Minho (Luis Pinto and colleagues), INRIA Sophia (Gilles Barthe), Leicester (Neil Ghani), Nottingham (Thorsten Altenkirch).