MoSIS/IKT
Modularity in large Software and Information Systems

The Programming Technology Group (PTG)
Department of Informatics, University of Bergen

October 9, 2001

Contents

1 Introduction

In its vision of change in the Norwegian industrial sector the Norwegian Ministry of Trade and Industry sees information technology as one of the important growth sectors [55]. To achieve this goal Norway will need practitioners highly skilled in the emerging software technologies. In accordance with this, the Research Council of Norway (NFR) has singled out software development methodologies as a prioretized area within its IKT (information and communication technology) research strategy.

The Programming Technology Group (PTG) at Department of Informatics, University of Bergen, is well positioned to contribute to these strategic goals. It has well-documented skills in software development methodology, the ability to take theory into practice, as well as an excellent record on education.

The research activities of the group are at a high international level and range from foundational work on specifications, logic and functional programming, to incremental process models for component technologies in software engineering.

Large software and information systems are among the most complicated of all artifacts. Although the human engineering genius has been able to assemble them, it is totally impossible to oversee the astronomical number of internal actions when executing such programs on a computer. It is even more difficult to oversee the range of all possible executions. The latter is nevertheless crucial for the proper functioning of those computer programs in many applications, and society relies more and more on computers for its security, prosperity and welfare. Experience has shown that confidence in the proper functioning of computer programs can only come from sharper understanding and can only be based on logically sound foundations. Given the sheer complexity of this problem, there is no other way than to apply techniques usually denominated as 'modularization' (separation of concerns) and 'abstraction' (reduction of details), techniques which have proved to be successful in many other engineering disciplines and in the natural sciences. Although most of these techniques are reasonably well understood on their own, their combination and interaction at various levels is not. This motivates the focus of the project on modularization in large, 'heterogeneous' software systems.

2 Goals

The overall goals of the MoSIS project are:

G1. Advancing the state-of-the-art of modularization techniques

- Modularization techniques aimed at substantial reduction of the cost and effort in development and maintainance of large software and information systems.
- A conceptual and formal framework for the composition and interaction of modules at various levels of abstraction in the development of robust and user-friendly software.

G2. Building up a national competence base around these issues
The aim is to strengthen the Norwegian academic community in the area of modular systems and system development.

G3. Disseminating and exchanging the knowledge acquired
The acquired knowledge is to be presented at appropriate national and international fora (journals, conferences, workshops).

G4. Transfer of existing and acquired knowledge to industry
Incorporation in courseware for the university curriculum. Employment in industry of M.Sc. and Ph.D. graduates produced.

3 Results

The following results are planned:

R1. A coherent theory of modular systems (goals G1, G2)
Confronted with a variety of module concepts and their use, we will review and organize these along one or more of the following dimensions:

- the notions of a module and its interface including basic pre- and postconditions, execution time and space properties, exceptions, etc.
- composition of modules, in particular, the means of relating modules with heterogenous interfaces and interacting behaviour
- different levels of abstraction as means of modularization in systems development

This result will provide the basis for the result **R2**.

R2. Advancing concepts and techniques for development and maintenance of modular system (goals G1, G2)
A set of concepts and techniques for developing and maintaining modular systems will be formulated on the basis of the theoretical results in **R1**.

R3. Scientific publications in the area (goal G3)
The results of the project will involve the standard scientific forms of presentation:

- papers in journals and at international conferences
- papers at national conferences, technical reports
- disclosure via the MoSIS project website.

We expect to produce around 15 publications during the project period.

R4. Training of experts in modularization technologies (goals G3, G4)
We expect to produce during the project period at least 6 M.Sc. and 2 Ph.D. students in this area. We will prepare courseware which will introduce students to the basic concepts. This includes B.Sc. students taking one or more courses related to the subject, but these numbers are difficult to quantify. Generally, the training will result in a large and lasting benefit by giving university students more relevant education in and exposure to novel software technologies.

R5. Bringing together industry and university on modularity (goals G3, G4)
Experts in new approaches pursuing a career in the industrial sector is one of the major transfer methods from academia to industry. Feedback from industry will influence both teaching and research.

4 Rationale

In this section we present the state-of-the-art with respect to modularization, the trends and needs in this area, as well as the position the PT-group takes in this field.

4.1 State-of-the-art

Modular decomposition of systems becomes very important as systems increase in size. The table below, which compares the sizes of some larger literary works with that of sample software systems developed in Bergen, indicate the need for controlling the complexity software.

Given the sheer complexity of large software/information systems, it is essential to modularize their design and implementation. In this way the software industry follows the path set out by

lines	kind of software	pages	length
750 000	logistics, oil platform	18 750	2 250m
426 000	subscription, newspaper	10 650	1 280m
300 000	loans/mortgages, insurance company	7 500	900m
80 000	data acquisition, industry	2 000	240m
30 000	quota regulations, government agency	750	90m
24 000	salaries, industry	600	72m
12 000	simple Pascal compiler	300	36m

lines	book	pages	length
73 000	J.R.R. Tolkien: *The Hobbit*, *The Ring Trilogy*, *The Silmarilion*	1 847	222m
26 000	Tom Wolfe: *The Bonfire of the Vanities*	661	132m

many other engineering disciplines, in particular by the industry producing micro-electronic hardware. Software, however, is produced in far greater variety and with very little standardization. We can distinguish various levels on which modularization takes place during the production of software.

- *Modular specification.* There are many practical attempts to support modularization in practical specification languages such as VDM [38, 27], Z [12] and UML [64], attempts which often lack theoretical foundation. On the theoretical side we mention algebraic specification of software (ASF [40], OBJ [30], Rewriting Logic [48, 56, 57, 58]), in particular parametrized algebraic specification [18, 19, 41, 42]. In the case of strongly typed formalisms ([11, 59], Coq [4]) we mention polymorphism as an important abstraction mechanism for modularization.

- *Software modules.* In this area one sees a whole range of techniques. Without any claim to be exhaustive we mention software components, packages, interfaces (such as in Java [31, 52]), libraries, as well as generic classes (templates, C++ [68]) and aspect [39] and feature [61, 20] oriented programming. This is moving the focus forward to **COTS** (components off the shelf), the increasingly important trading of software modules.

- *Development of modular systems.* Many software processes have been formulated for the development of modular software, see [67] for an overview. Recently object-oriented development processes, such as the unified process model [37] and extreme programming [5], have become commonplace. Our understanding of these processes has now matured enough that a theoretically founded understanding of modular software may provide valuable insights. Based on our work with algebraic software methodologies in the SAGA project we have proposed a 2-tiered software process [34, 33]:

  1. a domain analysis tier yielding specifications of domain concepts and domain architecture, and
  2. a tier of many independent development tasks for implementation of modules and application software, all falling within the domain architecture.

  In a recent study [34] we have shown that this software process model may reduce development costs of individual application programs by 90%. Algebraic techniques for system development [18, 14, 24, 15, 2, 32, 35] provide a rich conceptual framework facilitating such a process.

The Software Engineering Institute at Carnegie-Mellon University (USA) is one of the most influential research organisations on software processes and process improvements, recently with a focus on processes for COTS. Closer by are the NTNU (Trondheim) with for example the **INCO** project (project 5, section 9), covering the pragmatic aspects of system development, and the CWI (Amsterdam) with a project called **Coordination Languages** (project 6, section 9).

4.2 Trends and Needs

Modularization techniques are acknowledged to be a major factor in the successful development of large systems. This holds in particular for component-based technologies. Evidently, building software by composition of modules poses new problems [10]. At the same time our understanding of these phenomena is largely insufficient. Yet another trend concerns heterogeneous systems, for example for internet banking, where parts of a Cobol legacy have to be reconciliated with the newest Java applets. Developing a notion of a correct/sufficient definition/specification of a component and its interaction with the environment are of fundamental importance for the emergence and deployment of a component industry. This has a legal side, a pragmatic side (studied in e.g. the INCO project), as well as a formal/technical side which we propose to study here. A recent litigation case [21] involving component vendors underlines the importance of the technical issues.

Apart from a better understanding of the new challenges that the above trends entail, there is an urgent need to train students and employees at all levels in using the new techniques.

4.3 Position of the PT-group in the field of modularization

Reviewing the situation in the field of modularization we can distintinguish three approaches.

1. A pragmatic approach, which starts from existing languages and systems, but runs into difficulties by lack of theoretical underpinning.

2. A theoretical approach, usually based on one single, well-understood formalism, but not sufficiently expressive for the whole range of phenomena encountered in practice.

3. A broad-spectrum theoretical approach, which tries to combine the major contributions of the previous two approaches.

Approach 1 is pursued by the software industry and in R&D labs that are very close to production facilities. Approach 2 is to be found almost exclusively in academia. Approach 3 is the approach that we propose. It is seldom found in the software industry, since it is a long-term investment. Approach 3 is also seldom found in academia, since the results it will produce may be less elegant than with one single theory. We think that the practical importance of approaching modularity as under approach 3 outweighs by far a presumed lack of elegance, and that the difficulty and the long-term character call for a **focused research project** in academia. The results should, however, be transferred to industry as soon as possible, requiring experts trained in the new approaches.

The PT-group is particularly well-positioned to carry out this project. The members all have a thorough background in programming technology and its foundations. On the research level, the particular competences of the individual members cover a broad range, including most of the modularization and abstraction techniques employed in large software and information systems. (More details are given in section 7.)

Their scientific interests have long been directed towards the problems to be addressed by the proposed project:

- abstraction in description of software [7, 25, 24, 32, 34, 71, 72]

- compositionality and structuring of specifications through algebraic and type theoretic means [3, 19, 32, 34, 35, 73, 78, 79, 70, 74, 52]

- treatment of heterogeneity through abstraction and institution mappings [77, 46, 44, 36]

- abstract treatment of partiality [19, 81, 76, 36, 43, 41, 42]

The proposed project is thus a natural continuation of the earlier research and work of the PDT Group.

5 Work packages

W1. **Project management** This package covers project coordination and reporting.

(T1) **Task: Project coordination** There will be monthly meetings among the project participants for communication of the achieved results and planning.

Deliverables: Internal reports, work notes, drafts of reports and articles, presentations.

(T2) **Task: Project reporting**

Deliverables: Annual reports to NFR, status reports, web-updates.

W2. **Supervision package (result R4)**

(T1) **Task: Supervising Ph.D. 1**

(T2) **Task: Supervising Ph.D. 2**

(T3) **Task: M.Sc. supervision**

The supervision of M.Sc. students will be according the university procedures on this point. For every Ph.D. student there will be one senior staff member involved in the project made responsible for the supervision. Ph.D. supervision will take place at least once every two weeks. The supervision will be reviewed in the project meetings, if necessary.

W3. **Theory of modular systems (results R1, R2, R3, R5)**

(T1) **Task: Analysis of the existing theories of modularization**

Research challenges: Confronted with a variety of theories of modularization, these existing theories have to be reviewed and analysed along the following axes:

- the notions of a module and its interface
- composition of modules, in particular, the means of relating modules with heterogeneous interfaces
- different levels of abstraction as means of modularization in systems development

Deliverables: A series of studies describing the different theories of modularization and fixing the outcome of their analysis and comparison.

(T2) **Task: Coherent theory of modular systems**

Research challenges: Based on the analysis and comparison of different theories of modularization and the advances in existing theories in work package **W4**, a coherent theory of modular systems has to be developed. This includes an integrated view of the different theories, and of their adequate ranges of application.

Deliverables: Scientific papers in journals and conferences disseminating the basic scientific advances and results obtained by the project.

W4. **Defining component software (results R1, R2, R3, R5)**

(T1) **Task: Application of types to description of modules**

Research challenges: Software components are currently only weakly typed, there is seldom more than an informal partial description of the interface. Type theory is a powerful formalism that can be put to use to improve on this. Simple types can be used for signatures, subtypes for coercion of parameters, polymorphism can be expressed, and dependent types can formalize pre- and post-conditions and other properties of modules.

Deliverables: A type inference system and type checking algorithms for component software. Thus we obtain a framework for typing component software which is sufficiently expressive to capture a number of other abstraction techniques as well.

(T2) **Task: Algebraic description of modules**

Research challenges: Algebraic and coalgebraic specifications are used for defining the functionality of modules, but how to apply this in the definition of run-time resource usage and other non input-output oriented aspects of software is not well understood.

Deliverables: Papers on the extension of algebraic methods to non-functional aspects of modules descriptions.

W5. **Compositionality properties of modules (results R1, R2, R3, R5)**

(T1) **Task: Composition of heterogeneous modules**

Research challenges: Software components are important building blocks for software systems. The theoretical understanding of how various forms of components can be combined and how to control the effects of combining them needs to be better understood. Categorical semantics and algebraic specifications are key techniques here, but existing theory must be expanded and adapted for this purpose.

Deliverables: Paper(s) on compositionality, possibly with the guidelines for specifying and combining interfaces of modules.

W6. **Dissemination and contacts with industry and other partners (R3, R4, R5)**

The objective of this work package is to communicate the results obtained in work package **W3-W5** to the scientific community, as well as to transfer knowledge to industrial users.

(T1) **Task: Scientific presentation** The results of the project will be disseminated through the normal scientific channels.

Deliverables: papers in journals and presentations at national and international conferences.

(T2) **Task: Dissemination through teaching** Teaching of students at all levels will be an important form of transferring the gained knowledge to users outside academia. (Cf. work package **W2**.)

Deliverables: Two Ph.D. candidates, about 6 M.Sc. candidates, introduction of the basic concepts of modularization in the curriculum.

Work schedule

Preliminary timetable, given as project year/quarter, for the tasks and results of the project. The abbreviations refer to the work packages and tasks. When specific tasks have not been mentioned, then they have the same schedule as the work package.

year/quarter	2002				2003				2004				2005			
Work package	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
W1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
W2-T1	x	x	x	x	x	x	x	x	x	x	x					
W2-T2				x	x	x	x	x	x	x	x	x	x	x	x	
W2-T3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
W3	x	x	x	x	x	x	x	x								
W4			x	x	x	x	x	x	x	x	x	x	x	x	x	x
W5			x	x	x	x	x	x	x	x	x	x	x	x	x	x
W6			x	x	x	x	x	x	x	x	x	x	x	x	x	x

6 Resources

The project does not require extensive equipment or resources other than covering personnel and standard costs (PC style computer, travel and subsidence etc.). The planned costs, in NOK, based on project start January 2002 and 4 years duration, are given in the table below. Some flexibility in the various costs as described in the paragraphs below will be required.

Goal	2002	2003	2004	2005
Doctoral student	412.000	412.000	412.000	0
Doctoral student	177.000	412.000	412.000	235.000
Research associate	0	0	264.000	264.000
Running costs	500.000	500.000	500.000	500.000
Totals applied for	1.089.000	1.324.000	1.588.000	999.000
1/3 Full Prof. (funded by UiB)	250.000	250.000	250.000	250.000
3/3 Assoc.Prof. (funded by UiB)	600.000	600.000	600.000	600.000
Totals	1.939.000	2.174.000	2.438.000	1.849.000

6.1 New researchers

An important 'deliverable' will be new researchers receiving their doctorates on topics within the project's area. We have budgetted two positions for doctoral candidates, one starting January 2002, the other August 2002. We also budgetted 1 year funding for a *research associate* position, preferably at the post-doctoral level (depending on the recruitment possibilities, the scheduling in year 3,4 is tentative). Salary raises are included in the running costs.

It has been the long-standing policy of the group to expose doctoral candidates to the international research community through 6-12 months of funded stays at a cooperating institution abroad. The running costs therefore include a half year's stay abroad for each doctoral candidate (according to NFR funding for 'overseas fellowship' + travel).

6.2 Technical assistance

In order to maintain a WWW site on the MoSIS project and to help with case studies and other project support tasks we need technical assistance beyond what is reasonable to ask of Ph.D. students. We have therefore included a sum corresponding to 3 months work by a M.Sc. student every year.

6.3 International network

The PTG has hosted several researchers from abroad, on stays ranging from a few weeks to more than a year. It is important also in the future to be able to invite these, and other, highly qualified scientists for short to medium length visits. We should like to stress the importance of being able to fund visits *to* our university – it exposes students at *all* levels to the international research community. We budgetted annually two research visits of the partners at the University of Bergen, each stay to be of one month's duration.

Communication of achieved results to other scientists and critical reviews and feedback are necessary parts of all research. In computer science the primary feedback mechanism is through conferences and workshops rather than journal publications. It is therefore crucial that the project participants get the opportunity to present their results at international conferences and workshops and visit other research institutions for extended visits.

We budget an average of 10 conference travelsand 3 one month's visits to institutions abroad each year.All these are included under running costs.

7 The PT-Group

The group has the following members (their main research interests are given in parentheses):

- Professor Marc Bezem (type theory, logic programming, process algebras)
- Assoc. Professor Uwe Wolter (partial algebraic specifications, categorical semantics, relating alternative description frameworks, coalgebras, process algebras)

- Assoc. Professor Magne Haveraaen (algebraic software methodologies and frameworks, partiality, modular software in the problem domain of scientific computing)
- Assoc. Professor Khalid A. Mughal (object orientation, Java, web-based systems, distant learning)
- Assoc. Professor Michał Walicki (fundamentals of algebraic specifications, abstraction mechanisms in specification)

Professor Marc Bezem will function as the **Principal Investigator (PI)**. He is internationally recognized, both as (co-)author of over 30 refereed scientific publications ($\geq$ 250 citations in ResearchIndex) and as (co-)editor ([6, 22], 350 citations in ResearchIndex). He has been involved in many projects: PRISMA (parallel computing, Philips, 86–90), Integration (ESPRIT Basic Research Action, 88–90), Typed Lambda Calculus (Dutch Science Council, 90–94), Types for proofs and programs, Logical Frameworks (ESPRIT working groups, 90-ies). He is currently participating in ESPRIT Working Group 29001 TYPES (project 2 in section 9), presiding over the European Association for Computer Science Logic (97–02), and, e.g., invited lecturer on the EEF Summer School on Logical Methods (BRICS, June/July 01).

7.1 Research interests and quality

Given the group's academic profile in the area of components, heterogeneity and abstraction, the group has the required qualifications to establish competence in the areas of the proposed research.

The research of the group has been focused primarily on the algebraic, categorical and type-theoretical techniques for software development, with particular emphasis on the abstraction mechanisms needed for adequate description, modularisation and certification of specification and implementation. Their scientific interests and background reflect a long focus around the problems to be addressed by the proposed project (see also subsection 4.3).

The research interests of the group members, as well as the proposed project, fall very precisely within the area of the IKT programme: 'General software competence', with particular emphasis on stepwise, modular system development.

7.2 International contacts

The PTG has a wide range of international contacts and participates in several international projects, as was specifically recognized in the national evaluation of the Norwegian computer science research groups: *"The group [PTG] should be applauded for their international contacts"*[1]. Contacts and cooperation exist, in the form of joint publications and supervision of students with researchers at the universities at Swansea, Munich, Utrecht, Vilnius, Warsaw and with CWI in Amsterdam. Contacts also extend to SRI International and Technical University of Berlin.

Because of these contacts the group has been able to attract first rate international researchers to positions – M. Bezem and U. Wolter, who joined the group in 2000, are internationally recognized researchers who moved to Bergen from, respectively, universities in Utrecht and Berlin.

The international contacts are reflected in participation by the group members in a series of international projects:

- **Types** – ESPRIT Working Group 29001 in the IST programme, which forms the european platform for research on **typing systems**.
- **COMPASS** – ESPRIT Basic Research Working Group which has brought together most european scientists in the area of **algebraic specification methods** in an effort to consolidate and integrate the theoretical basis and apply it to **software technology**. (1989–1996)[14, 2]

[1] "Informatikk: Research and teaching in Norway: a critical evaluation", NAVF – The Council for Natural Science Research 1992.

- **CoFI** – european initiative, started in 1995, to provide a common framework for **algebraic specification and development** and including the design and implementation of a general-purpose algebraic specification language. (EU funding 1998–2001)[14]
- **SAGA** – started as an ESPRIT project with partners from CWI (NL) and University of Wales Swansea (UK). Now continues as an NFR-supported project with collaborators also at university of Utrecht (NL) and Wagner Mathematics (USA). SAGA applies algebraic software methodologies to the analysis and design of large and distributed software systems.
- **Architectural Abstraction** – NFR funded project concerned with **abstraction** in description and design **software components**. (completed in 2000)
- **KORSO** – german project including 15 groups from academia and industry devoted to combine and to integrate formal based techniques and methodologies for developing **correct software**. (1991–1994)[15]

As a part of this project, the group will continue and strengthen the cooperation with other international collaborations, in particular:

- Professor Sigurd Meldal (from CalPoly and Stanford University, USA)
- Dr. Eric Wagner (from Wagner Mathematics, USA)
- Dr. Alfio Martini (from UFRGS, Porto Alegre, Brasil)
- Professor Henk Barendregt (Nijmegen University, NL, and Carnegie Mellon University, USA)
- Professor Jaco de Bakker (CWI and Free University, Amsterdam, NL)

8 Industrial and societal relevance

According to a recent report [62] there are around 2900 people employed in the IKT-industry in the Bergen region. There is a need for 2000 more professionals in the next 3 years. Nearly 80% of the companies believe that the regional educational institutions are fertile ground for recruitment. The report also shows that only 15% of these companies have any collaboration with the local research institutions. The report recommends the public authorities to encourage and establish a climate for greater cooperation between interested parties. Both the University and the Bergen College are seen as major resources to draw upon. The MoSIS project will contribute significantly to meeting the strategic needs of IKT-industry in the Bergen area.

Future IKT-investments in the Bergen area include a radical expansion of the Bergen Advanced Technology Centre (HIB) and the construction of a Science Park near Bergen Airport. The success of these plans depends on the educational institutions supplying the qualified personnel. This stresses once more the strategic aspects of the proposed research.

The PT-group is one of the most prolific research groups at the Department of Informatics when it comes to producing M.Sc. students in Informatics (≥ 30 over the years 93–00, with a staff of only 3 permanent faculty members). The group runs a web-based course on Object-oriented programming in Java. Further collaboration on this point with Bergen College is envisaged via the MediaLab. In addition it has produced several doctorates in the field (most recently P. Ölveczky, December 2000) and expects another doctoral candidate to complete his degree in the near future (Y. Lamo in 2002). Two additional doctoral candidates would join the group within the MoSIS project. These graduates are eagerly sought after by the IKT-industry in the local and regional areas. The group members also continue to have on-going research collaboration with various companies on the West coast of Norway. The group has also successfully concluded co-supervision of several M.Sc. students in collaboration with the IKT-industry where the problem-domain has been defined by the external partner and the group members have contributed with their expertise. The group members have also given courses/seminars for the IKT-industry and are often called upon by the IKT companies for consultation on IKT policy matters.

In the table below, an overview of the group's involvement in the local and the regional scene is presented within specific areas of IKT, the nature of the collaboration and identification of external partners/contacts.

IKT area:	**Industrial partner/contact:**
Telecommunication Systems	Protek Telesoft Nera Telenor
Marine Information Systems	NERSC Nansen Environmental and Remote Sensing Center
Embedded systems/ Component-based computing/ Oil industry	ICESOFT/ WIND RIVER Seven Mountains, UniGEO Globe Systems, Rogaland Forskning CMR (Christian Michelsen Research)
Banking and Insurance	DNB, Vital New Technology Consulting AS Telenor/Novit
eBusiness	Ementor, Optec

9 Related projects

The proposed project is related to several other activities and projects involving the members of PTG and their partners/competitors. Although the specific goals may differ they are all closely related to the interests motivating the research of PTG. Consequently, all these activities will support and benefit from each other.

1. **HyperEducator.** In [54] Mughal and Kroepelien of the University of Bergen, propose a 4-phase approach to the construction of a flexible, generic system for the development of e-learning content. This approach encompasses the whole process from course content (e.g. a lecture on Java) to a web-based presentation. This development process certainly requires a set of tools, the design, implementation and maintenance of which comes with all the difficulties of large software systems in general.

2. **TYPES** is EU Working Group 29001 in the Information Societies Technology programme (2000–2002, `www.durham.ac.uk/TYPES`). The subject of this project is Computer-Assisted Formal Reasoning, an area which is of deep interest for industry. The aim of our research activities is to develop the technology of formal reasoning based on Type Theory by improving the languages and tools of formal reasoning and by applying the technology in several domains such as programming languages, certified software, and formalisation of mathematics. TYPES is based on the strong collaboration and achievements in three successful European projects (ESPRIT BRA and Working Group), in which we have built several computer systems for proof development and used them in applications.

3. **CoFI** – the Common Framework Initiative – aims at designing a common framework for algebraic specifications. The working group has received funding from the EU and involves most of the researchers in the area of algebraic specification technology. CoFI aims at designing a unifying family of algebraic specification languages, whose kernel language, CASL, is at the final stage of development.

4. **SAGA** – Scientific Computing and Algebraic Abstractions – is a project coordinated by M. Haveraaen where algebraic methodologies play an important role in the development of application software. This project involves partners in Wales (University of Wales Swansea) and the Netherlands (CWI and University of Utrecht), has previously received ESPRIT funding. Currently NFR is funding SAGA-II which also involves cooperation with the University of Utrecht and Wagner Mathematics.

5. **INCO** – Incremental and Component-Based Software Development – is a project at NTNU (Trondheim) and the University of Oslo (2001–2003, `www.uio.no/~isu/INCO`). It aims at advancing the state-of-the-art as well as the-state-of the-practice of software engineering, focusing on technologies for incremental and component-based software development. The approach is to propose, refine and validate enhanced technologies. The focus is pragmatic: overall planning, risk estimation, cost/risk ratio. The methods are empirical rather than formal.

6. **Coordination Languages** is a project at CWI (Amsterdam, NL) under the direction of J. Rutten and F. Arbab. The project is focused on the development of formal models for coordination, components, and component based software. An important activity in this project is the development and application of the system Manifold, a coordination language for orchestration of the communications among independent, cooperating processes in a massively parallel or distributed application.

10 References

[1] E. Astesiano and M. Cerioli. Relationships between logical frameworks. In M. Bidoit and C. Choppy, editors, *Recent Trends in Data Type Specification. Dourdan France 1991*, pages 126–143. Springer LNCS 655, 1993.

[2] Astesiano, E., Kreowski, H.-J., Krieg-Brückner, B.(eds.): *Algebraic Foundations of System Specification*, 1999.

[3] E. Barendsen and M.A. Bezem. Polymorphic extensions of simple type structures, with an application to a bar recursive minimization. *Annals of Pure and Applied Logic*, 79(3):221–280, 1996.

[4] B. Barras, S. Boutin, C. Cornes, J. Courant, J.-C. Filliâtre, E. Giménez, H. Herbelin, G. Huet, C. Muñez, C. Murthy, C. Parent, C. Paulin-Mohring, A. Saïbi, B. Werner. *The Coq Proof Assistant Reference Manual, version 6.2.4.* INRIA, 1998.

[5] K. Beck: Embracing change with extreme programming. *IEEE Computer* 32(10), 1999, pp. 70-78.

[6] M.A. Bezem and J.F. Groote, editors. *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science 664. Springer-Verlag, March 1993.

[7] M.A. Bezem, R.Bol and J.F.Groote. Formalizing process algebraic verifications in the Calculus of Constructions. *Formal Aspects of Computing*, 9:1–48, 1997.

[8] M.A. Bezem. An improved extensionality criterion for higher-order logic programs. *Proceedings CSL01*, to appear.

[9] B. Boehm: A spiral model of software development and enhancement. *IEEE Computer* 21(5), 1988, pp. 61-72.

[10] B. Boehm. COTS integration: plug and pray. *IEEE Computer* 32(1), 1999, pp. 135-38

[11] L. Cardelli and P. Wegner. On understanding types, data abstraction, and polymorphism. *Computing Surveys*, vol. 4, 1985, pp. 471–522.

[12] J.P. Bowen and A. Fett and M.G. Hinchey. *ZUM'98: The Z Formal Specification Notation.* Springer-Verlag, LNCS 1493, 1998.

[13] M. Cerioli. *Relationships between Logical Formalism.* PhD thesis, Università de Pisa–Genova–Udine, 1993.

[14] Cerioli, M., Gogolla, M., Kirchner, H., Krieg-Brückner, B., Qian, Z., Wolf, M. (eds.): *Algebraic System Specification and Development - A Survey and Annotated Bibliography*. Shaker Verlag, 1998.

[15] Broy, M., Jähnichen. (eds.): *KORSO: Methods, Languages, and Tools for the construction of Correct Software*. Springer LNCS 1009, 1995.

[16] M. Cerioli and J. Meseguer. May I Borrow Your Logic? In A. M. Borzyszkowski and S. Sokolowski (eds.), *Proceedings of the 18-th International Conference on the Mathematical Foundations of Computer Science*. Springer LNCS 711, 1993.

[17] M. Cerioli, T. Mossakowski: From total equational to partial first order, *Chapter 3* in [2].

[18] I. Claßen, H. Ehrig, D. Wolz: *Algebraic Specification Techniques and Tools for Software Development*. AMAST Series in Computing, Vol. 1. World Scientific Publishing, 1993.

[19] I. Claßen, M. Große-Rhode, U. Wolter: Categorical concepts for parametrerized partial specifications, *Mathematical Structures in Computer Science*, 5, 2, 1995.

[20] K. Czarnecki and U. Eisenecker. Synthesizing Objects. *Journal of Concurrency: Practice and Experience*, 2000.

[21] Dagens Næringsliv, 30 Mai 2001, p. 53.

[22] D. van Dalen and M.A. Bezem, editors. *Computer Science Logic '96*, Lecture Notes in Computer Science 1258. Springer-Verlag, April 1997.

[23] R. Diaconescu and K. Futatsugi. *CafeOBJ report: the language, proof techniques and methodologies for object-oriented algebraic specification*, volume 6 of *AMAST series in computing*. World Scientific; Singapore, 1998.

[24] H. Ehrig, M. Gajewsky, and U. Wolter. From abstract data types to algebraic development techniques: A shift of paradigms. In F. P. Presicce, editor, *Recent Trends in Algebraic Development Techniques*, volume 1376 of *Lecture Notes in Computer Science*, pages 1–17. Springer Verlag, 1998.

[25] H. Ehrig, M. Große-Rhode, U. Wolter: Applications of category theory to the area of algebraic specification in computer science, *Applied Categorical Structures*, 6, 1998.

[26] H. Ehrig, B. Mahr: *Fundamentals of Algebraic Specification*, Springer, 1985.

[27] J. A. Goguen. An Algebraic Approach to Refinement. In: *VDM 90: VDM and Z — Formal Methods in Software Development. 3rd International Symposium of VDM Europe*, Springer LNCS 428, 1990, pp. 12–28.

[28] J. Goguen, R. Burstall: Introducing Institutions, in *Logic of Programs*, LNCS vol. 164, Springer, 1983.

[29] J. Goguen, R. Burstall: Institutions: abstract model theory for specification and programming, in *Journal of the ACM*, vol. 39, no. 1, 1992.

[30] J. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, J.P. Jouannaud: Introducing OBJ, in J.A. Goguen and G. Malcom (eds.) *Software Engineering with OBJ: Algebraic Specification in Practice*, Cambridge University Press, 1996.

[31] J. Gosling, B. Joy, and G. Steele. *The Java Language Specification*. Addison-Wesley, 1996.

[32] M. Haveraaen. Case Study on Algebraic Software Methodologies for Scientific Computing. *Scientific Programming*, 8(4), 2001.

[33] M. Haveraaen. A 2-tiered software process model for utilizing CASL. Technical Report 208, Department of informatics, University of Bergen, P.O.Box 7800, N-5020 BERGEN, Norway, October 2000.

[34] M. Haveraaen, H. A. Friis, and T. A. Johansen. Formal software engineering for computational modeling. *Nordic Journal of Computing*, 6(3):241–270, 1999.

[35] M. Haveraaen, V. Madsen, and H. Munthe-Kaas. Algebraic programming technology for partial differential equations. In A. Maus, F. Eliassen, et al., editors, *Norsk Informatikk Konferanse – NIK'92*, pages 55–68. Tapir, Norway, 1992.

[36] M. Haveraaen and E. G. Wagner. Guarded algebras: Disguising partiality so you won't know whether its there. In D. Bert and C. Choppy, editors, *Recent Trends in Algebraic Development Techniques*, volume 1827 of *Lecture Notes in Computer Science*. Springer Verlag, 2000.

[37] I. Jacobsen, G. Booch, J. Rumbaugh. *The unified software development process*. Addison Wesley, 1998.

[38] C.B. Jones. *Systematic Software Development Using VDM*. Second edition, Prentice Hall International, 1995.

[39] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. V. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-Oriented Programming. In: M. Aksit and S. Matsuoka (eds.), *Proceedings of 11th European Conference on Object-Oriented Programming* (ECOOP '97), pp. 220–242. Springer-Verlag 1997.

[40] P. Klint. A meta-environment for generating program environments. *ACM transactions on Software Engineering and Methodology* 2(2):176–201, 1993.

[41] Lamo, Y., Walicki, M. *Specification of Parameterized Programs – Persistency Revisited*, to appear in *Nordic Journal of Computing*, (2001).

[42] Lamo, Y., Walicki, M. *Composition and Refinement of Parameterized Data Type Specifications*, Proceedings WADT'2001, Genova, Springer (to appear).

[43] Lamo, Y., Walicki, M. *Modeling partiality by nondeterminism*, Proceedings of The Fifth Multi-Conference on Systemics, Cybernetics and Informatics, SCI2001/ISAS2001, Orlando FL, USA, 2001.

[44] Y. Lamo, M. Walicki: Structuring Specifications and Software using Nondeterminism – forthcoming.

[45] D. Luckham, S. Meldal, J. Vera: Three concepts of system architecture, in *Foundations of component based systems*, Oxford University Press [to appear]

[46] A. Martini, U. Wolter: A systematic study of mappings between institutions, in *Recent Trends in Data Type Specification*, LNCS vol. 1376, Springer, 1997.

[47] S. Meldal, D. Luckham: NSA's MISSI Reference Architecture – from prose to specification, in *Requirements Targeting in Software and Systems Engineering*, LNCS vol. 1526, 1998.

[48] J. Meseguer: *Conditional rewriting logic as a unified model of concurrency*, *TCS*, 96, 1992.

[49] J. Meseguer. General logics. In *Proc. Logic Colloquium '87*. North-Holland, 1989.

[50] T. Mossakowski: Equivalences among various logical frameworks of partial algebras, in *Proc. CSL'95*, LNCS vol. 1092, 1995.

[51] T. Mossakowski: Different types of arrows between logical frameworks, in *Proc. of ICALP'96*, LNCS vol. 1099, 1996.

[52] K. A. Mughal and R. W. Rasmussen. *A Programmer's Guide to Java Certification*. Addison-Wesley; Reading, Mass, 2000.

[53] K. A. Mughal and S. Nyster and E. Haagensen. Bergen Webucator Web-based tools for distributed learning. Electronic proceeding of the EADTU (European Association of Distance Teaching Universities). Paris Millennium Conference. 28. - 30. September, 2000.

[54] K.A. Mughal and B. Kroepelien. HyperEducator, a hypermedia system for the development of IKT-based learning. NFR proposal, 2001.

[55] NHDept. *Strategiplan for Nærings- og handelsdepartetementet*, May 2001, `http://odin.dep.no/nhd/norsk/dep/om_dep/strategi/index-b-n-a.html`

[56] P. Ölveczky, J. Meseguer: Specifying Real-Time Systems in Rewriting Logic. In J. Meseguer (ed.), *Proceedings of the 1st International Workshop on Rewriting Logic and its Applications*, Pacific Grove, 1996. Vol. 4 of *Electronic Notes in Theoretical Computer Science*, Elsevier 1996.

[57] P. Ölveczky, S. Meldal: Specification and Prototyping of Network Protocols in Rewriting Logic, *Proc. NIK'98*.

[58] P. Ölveczky, J. Meseguer: Specification of real-time and hybrid systems in rewriting logic, [submitted to *Theoretical Computer Science*].

[59] J. Palsberg and M.I. Schwarzbach. *Object-Oriented Type Systems*. John Wiley & Sons Ltd., Chichester, UK, 1994.

[60] J. Power. *Institutional Approach to Programming Language Specification*. Ph.D. Thesis, School of Computer Applications, Dublin City University, 1994.

[61] C. Prehofer. Feature-Oriented Programming: A Fresh Look at Objects. In: M. Aksit and S. Matsuoka (eds.), *Proceedings of 11th European Conference on Object-Oriented Programming* (ECOOP '97), pp. 419–443. Springer-Verlag 1997.

[62] Proteus Consulting. Bergen som IT-by? *Grov kartlegging av IT-miljøer i Bergen*, study carried out for the Bergen City Council.

[63] H. Reichel: *Initiality, Computability, Algebraic Specifications and Partial Algebras*, Oxford Science Publications, 1987.

[64] J. Rumbaugh and I. Jacobson and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1998.

[65] D. Sannella and A. Tarlecki. *Foundations of Algebraic Specifications and Formal Program Development*. Cambridge University Press. To appear.

[66] D. T. Sannella and A. Tarlecki. Extended ML: an institution-independent framework for formal program development. In *Proc. Workshop on Category Theory and Computer Programming*, pages 364–389. Springer LNCS 240, 1986.

[67] I. Sommerville. Software process models. In: A. Tucker (ed.) *Handbook of computer science and engineering*, CRC press, 1997.

[68] B. Stroustrup. *The C++ Programming Language (Special Edition)*. Addison Wesley. Reading Mass. USA. 2000.

[69] A. Tarlecki. Moving between Logical Systems. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specification*, volume 1130 of *Lecture Notes in Computer Science*, pages 478–502. Springer Verlag, 1996.

[70] M. Walicki, M. Broy. Structured specifications and implementation of nondeterministic data types. *Nordic Journal of Computing*, vol. 2, 1995.

[71] M. Walicki, S. Meldal. Multialgebras, Power Algebras and Complete Calculii of Identities and Inclusions. In E. Astesiano, G. Reggio, A. Tarlecki, editors, *Recent Trends in Data Type Specification*, volume 906 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.

[72] M. Walicki, S. Meldal. A Complete Calculus for the Multialgebraic and Functional Semantics of Nondeterminism. *ACM ToPLaS*, vol. 17, no. 2, 1995.

[73] M. Walicki, M. Haveraaen, S. Meldal: *Computation Algebras. Mathematical Structures in Computer Science*, 11(5):597–636, 2001.

[74] M.A. Walicki, S. Meldal: Singular and plural nondeterministic parameters, *SIAM J. of Computing*, vol. 26, no. 3, 1997.

[75] M.A. Weiss *Data Structures and Algorithm Analysis in Java*, Addison-Wesley, 1999.

[76] U. Wolter: An algebraic approach to deduction in equational partial Horn theories, *J. Inf. Process. Cybern. EIK*, 27, 2, 1990

[77] U. Wolter, A. Martini: Shedding light in the world of logical systems, in *CTCS'97*, LNCS vol. 1290, 1997.

[78] U. Wolter: Institutional Frames, in *Recent Trends in Data Type Specification*, LNCS vol. 906, 1995.

[79] U. Wolter, R. Wessäly, M. Klar, and F. Cornelius. Four Institutions. Bericht Nr. 94-24, Technische Universität Berlin, 1994.

[80] U. Wolter. A coalgebraic introduction to CSP. *Proc. of the 2nd Workshop on Coalgebraic Methods in Computer Science*, ETAPS'99, March 20–21 1999, Amsterdam, The Netherlands.

[81] U. Wolter. CSP, Partial Automata, and Coalgebras. *Theoretical Computer Science*, to appear.