IST-2001-33149 (MRG)

Periodic progress report: year 3

17th March 2005

Project start date: 1 Jan 2002
Project duration: 40 months
Project coordinator: University of Edinburgh
Project partners: University of Edinburgh, Ludwig-Maximilians-Universität München

Funded by the European Community's "Information Society Technologies" Programme (1998–2002) under the FET proactive initiative on Global Computing.

Contents

1 Executive Summary

The MRG project started on 1st Jan 2002. The aim of the project is to develop the infrastructure needed to endow mobile code with independently verifiable certificates describing its resource behaviour. These certificates will be condensed and formalised mathematical proofs of a resource-related property which are by their very nature self-evident and unforgeable.

Following the recommendation of the reviewers in the first review to simplify the MRG workplan, we submitted a list of proposed changes in last year's Periodic Progress Report. After the reviewers indicated their approval we submitted a revised version of the workplan reflecting these proposals, dated 8 October 2004. The number of workpackages remains unchanged but the number of tasks and deliverables is reduced: some related ones are combined, and some that are peripheral to the core objectives of the project are eliminated. At the same time, a few new tasks were added that are relevant in view of developments during the first two years of the project. A four-month extension was agreed so that the project would still be running at the time of the planned final review.

The MRG workplan is structured into ten technical workpackages, plus one administrative workpackage. Five of these were scheduled for activity during year 3. After year 2 most of the core components of the planned MRG infrastructure were in place and so some of the scheduled work for year 3 concerns enhancements and refinements.

We continued to take advantage of underspend on personnel in Edinburgh and the availability of well-qualified researchers to recruit additional manpower at that site, and this enabled good progress to be made in all areas. Only a few technical deliverables were actually completed during year 3, but major progress was made on nearly all that remain with completion of most planned during the first two months of the extension period.

The deliverables for year 3 are in the form of reports or software prototypes. All reports and software are downloadable from the project website. The content of the deliverables and key decisions are explained in Section 2.2 below, and Deliverable D11i compares progress with checkpoints and quality measures previously specified in the Self Evaluation Plan (D11f). Six deliverables produced between 31st Dec 2003 and the date of the previous Periodic Progress Report (23rd February 2004) are not discussed since they were already reported there. However, the work done on these in the period 1st Jan – 23rd Feb 2004 is reflected in the figures on manpower, which cover the entire calendar year.

2 Work progress overview

2.1 Specific objectives for the reporting period

As discussed during the second review meeting, our main goal during year 3 was to complete the core components of the planned MRG infrastructure to give a full implementation of proof-carrying code for heap space bounds, and to conduct a number of improvements. The core components in question are the following:

- Complete work on heap space type inference for Camelot (task 7d)
- Extend work on the bytecode logic (deliverable D2a/b) to yield a component that will generate heap space bound certificates from space types (task 6a)

Space type inference for Camelot was already implemented by the end of year 2, but certificate generation was a major task that consumed substantial resources during year 3. Additional priorities during year 3 were other tasks from WP7 and scheduled tasks from WP8 and WP9.

Section 2.2 and the final self-assessment report (D11i) give an overview of the work done towards these objectives.

2.2 Overview of the progress of the project during the reporting period

The following table lists the deliverables that have been completed since the last report.

Del. no.	Deliverable name	WP no.	Est. person-months	Del. type	Planned delivery (month)	Complete?
D4e/f	Soundness of type system	4	11	report	4/04	5/04
D6a	Certificate generator	6	27.5	prototype	5/04	12/04
D7a	Extension of type system by allowing user annotations [optional task]	7	4	report	6/04	no
D7c	Extension of basic type system to object-oriented language	7	6	prototype, report	9/04	no
D7d	Methods to infer type annotations automatically	7	10	report, maybe prototype	5/04	no
D7e	Extension of basic type system to mutable state and concurrency	7	2	report	9/04	no
D8a	Relationship between proof-based certificates and present-day security management	8	6	report	10/04	no
D8d	Methods for estimating cost of native methods	8	2	report	8/04	no
D9b/c	Advanced techniques of certification	9	2.5	report	12/04	no

Workpackage 4 is thereby completed. Because the remaining tasks in Workpackages 5 and 10 were eliminated in the revised workplan, these workpackages are also now complete.

Below is a verbatim copy of relevant parts of the MRG workplan, augmented with a description of what work has been carried out under each workpackage and task and explaining significant departures from what was foreseen. Workpackages and tasks on which work was neither planned nor carried out during year 3 are not included.

WP4: From reasoning principles to high-level type systems

Objectives: Develop reasoning principles and type systems for characterising resource usage, including a typechecker and soundness proofs.

This workpackage builds on the foundational strands in the first three packages. Beginning from the experimental high-level language designed in 3a, we investigate ways of expressing resource constraints and proving that they are satisfied by the compiled program, according to the cost model of 1b. We begin from reasoning principles, perhaps related to the bytecode logic, and then move towards type systems. The notion of type is a very broad one: type-checking can be used to

enforce simple consistency checks (that the addition operation + is always applied to two numeric arguments) but also rich semantic notions (such as interference [Rey78], presence of side-effects [TJ92], and resource usage as proposed here or studied in [Hof00b]).

Our approach is to stick with type systems where the type-checking problem is *decidable*, whereas the problem of proving that a resource constraint is satisfied will generally be undecidable. This means that we accept an unavoidable gap (the "slack") between the set of programs which are typable in a resource type system and the larger set of programs which satisfy the resource property of interest. For many natural examples, though, the resource bounds are met for obvious reasons which are in the scope of our type systems. The craft of designing type systems lies in capturing these natural examples and minimising the slack, while retaining a practical notion of type and practical type-checking algorithms.

e/f. Prove **soundness of type system**. *(11 months)*
Deliverables: technical report, conference/journal paper *Prerequisites: 4b/c*
It is natural to ask that our type systems should bear a close relationship to the cost model, so that they can be understood intuitively. Soundness is fundamental: a typing assertion should imply the intended cost constraints, as expressed by our model. Since the cost model applies to the byte code, we must reason about the translations made by the Compiler implemented in 3b. We will conduct a detailed pencil and paper proof to validate this claim.

Our type system should also be sound with respect to the bytecode logic for the compiled byte code. In other words, a typing assertion should imply a corresponding proposition in the bytecode logic. We will prove that. It is also something of a test for the bytecode logic, since that is expressed in rather different terms than the type system. Therefore insights from the type system development may lead to tweaks in the proof rules for the bytecode logic, although in general the latter should be stronger (i.e. capable of proving more).

Work carried out: Completed and delivered in May 2004 as Deliverable D4e/f. Rather than provide an informal argument that the space type system is sound with respect to the bytecode logic, we have worked on formally embedding the type system into the logic, by using a form of "modified assertions" constructed on top of the bytecode logic. The modified assertions are closely related to the type system and its soundness proof (albeit presented at the level of Grail code rather than Camelot).

g. **Milestone: Soundness proofs.** *Prerequisites: 4e/f*
This has been successfully achieved by the delivery of Deliverable D4e/f.

The research in this package is a combination of improving existing work and combining several different ideas, previously treated in separation and with different motivations. Drawing together these strands is a vital step towards our vision.

WP6: Generation of certificates

Objectives: Define format of certificates and implement a certificate generator. Experiment with reducing size of certificates.

This package is concerned with generating mobile guarantees of resource boundedness. The guarantee, or *certificate*, is what will be shipped together with the code, as irrefutable evidence for the consumer that the code obeys the desired resource constraints. Here we consider the format of the certificates, and their generation.

a. Implement a **certificate generator**. *(27.5 months)*
Deliverables: prototype implementation *Prerequisites: 4b/c*
Given a program which is typed in one of the high-level type systems developed in 4 and 5, we want to automatically generate a certificate which provides manifest evidence of this fact. The certificate contains a proof in our program logic. Here we must design a format for certificates (perhaps based on XML), and implement a software component which generates these from type-checked programs.

Work carried out: Completed and delivered in December 2004 as Deliverable D6a. The deliverable does not explicitly describe a textual or binary format for certificates (which is theoretically uninteresting), but in Section 5 it describes their semantic content. Specifically, a certificate contains the results of the resource type inference, encoded as a method specification table that associates one formula (a "derived assertion" in the sense of D6a) to each method of the program, together with some program information that eliminates the need to perform proof search. This information is stored in an Isabelle proof script produced by the compiler. The Isabelle representation of the program is generated by the consumer, which issues calls to the verification tactic (one for each method declaration) and combines the results to complete the formal proof.

a. **Milestone: Certificate generator.** *Prerequisites: 6a*

This has been successfully achieved by the delivery of Deliverable D6a.

The work in this package is mainly applied research, involving the design and construction of software components. Working software from this package will be an essential part of the final overall system.

Package 9 is dedicated to advanced research topics with the motivation of reducing certificate size. Parts 6c and 6d in this workpackage address this issue. We expect these approaches to be fruitful but not the final word; the results will be useful input for 9.

WP7: Advances in high-level type systems

Objectives: Improve expressiveness, user-friendliness, and accuracy of the type systems developed in WP4 and WP5.

The goal of this workpackage is to improve expressiveness, user-friendliness, accuracy of the type systems developed under 4 and 5.

a. Optional: Improve accuracy by **allowing for user interaction**. *(4 months)*
Deliverables: technical report *Prerequisites: 4a,4b/c,5a/b*
The aim here is to augment the basic type system from 4 with user annotations in the form of supplied proofs based, for example, on the derived rules from 4a. Also more abstract annotations such as loop invariants could be considered here.

If very restrictive resource bounds are imposed (e.g. hard limits on stack size) we might have to give the programmer the possibility to implement certain critical methods directly in bytecode with corresponding certificates obtained by hand, supported by a theorem prover. The task here will be to enable smooth integration of such user-supplied and -certified routines with high-level code.

Work carried out: When the MRG workplan was originally drawn up it was anticipated that it would be possible to devise type systems which would facilitate reasoning about heap

usage and to design typechecking algorithms for these systems. However, it seemed likely that type *inference* (as opposed to typechecking) for such systems would be difficult and that it would be necessary to extend the type system by allowing the programmer to provide hand-written type annotations describing resource usage: this extension was to form the basis of D7a. In fact, it turned out that automatic type inference for our resource-aware type systems was entirely feasible. We thus consider that the requirements behind this task are satisfied by work on 7d below, and have nearly finished a combined deliverable D7a/d.

c. **Extend** basic type system **to object-oriented language** *(6 months)*
Deliverables: research paper; extension of implementation *Prerequisites: 3d,4b/c,5a/b*
The basic type system developed in 4 will be likely to encompass only the first-order side effect-free fragment of our high-level language. The aim here is to generalise to account for object-orientation. While we may be able to draw inspiration from encodings of object-oriented languages in functional ones [BCP97], genuine innovation is required for two reasons. Firstly, these encodings invariable rely on advanced features such as higher-order functions and (mixed-variance) recursive datatypes; secondly the compilation of object-oriented features will make use of the object-oriented structure of the VM rather than following an encoding. Our strategy will be to extend the previous work to encompass higher-order functions (perhaps restricted to linear [Wad90, Hof00a]) and then tackle object-orientation proper.

Another issue that might arise would be an extension of the previous work to more object-specific resources such as "number of class and interface loads required".

Work carried out: Hofmann and Jost have begun to work on this topic recently. Good progress has been made but the topic has been found to be more interesting than anticipated: moving to an object-oriented target opens the way for making much more far-reaching improvements and extensions than expected. Additional effort will be required to complete this task during the final four months of the project.

d. **Type inference** *(10 months)*
Deliverables: research paper; implementation (optional) *Prerequisites: 4b/c,5a/b,7a*
The basic type system developed under 4 and the extension developed under 7a may rely on any number of user-supplied type annotations, for instance, recursive functions might be annotated by suggested bounds on their resource usage which are merely certified, cf. [CW00].

The aim here is to develop ways to infer such annotations automatically to a certain degree based on decision procedures for arithmetic inequalities [HP99], automata-theoretic methods [PWO97], unification [Mil78], program analyses, fixpoint methods [PS91], etc.

Work carried out: Automatic type inference analysis has assumed a more central role than we originally anticipated, and we have made significant research advances on this topic. Type inference following [HJ03b] is now a part of the Camelot compiler, and there is also a sophisticated automatic analysis of sharing as described in [Kon03]. Both of these analyses are used to provide information to the certificate generation phase in WP6. We have nearly finished Deliverable D7a/d that reports on this work.

e. **Extend** basic type system **to mutable state and concurrency** *(2 months)*
Deliverables: technical report *Prerequisites: 3f*
We will merely assess and elaborate the requirements for this extension possibly leading to future work.

Work carried out: Remarks on types for the concurrent extensions to Camelot presented in D3f will be included in Deliverable D7e, which will be ready for delivery in the near future.

This workpackage forms part of the scientific core of the proposal. Successful completion will demonstrate that our proposal extends beyond the basic feasibility validated in 1, 2, 4. The goals set out here are ambitious but realistic.

WP8: Integration with existing security model

Objectives: Implement resource manager and relate proof-checking infrastructure to present-day security management.

The preceding workpackages have detailed a proof-checking infrastructure which advances the state of the art in security management capabilities. This workpackage will enrich our understanding of this infrastructure by relating it to present-day security management.

a. **Relationship with existing security management** *(6 months)*
Deliverables: technical report *Prerequisites: 6a*

One direct advantage of a security infrastructure based on certificates and proof would be the ability to safely disengage the existing security manager for any downloaded code which can be shown not to offer any potential threat to the host cf. [Gil00]. Thus a *dynamic* (run-time) security management overhead would be replaced by a *static* (load-time) security assessment cost for those mobile code routines whose certificate guarantees that it satisfies the resource requirements of the host. Where the attached certificate *cannot* provide this guarantee (or cannot *be shown to* provide this guarantee) then either all or some parts of the existing dynamic security management code will be needed to supplement the static security assessment.

Work carried out: Work on integrating the proof checker of D2c with the J2SE5.0 Java Virtual Machine via "hooks" in the JVM known as "agents" has been completed. This is being written up for publication and will be ready for delivery in early 2005.

d. **Resource typing of native methods** *(2 months)*
Deliverables: operational techniques *Prerequisites: 1b*

It will be necessary to have a least a conservative estimate of the likely cost of invoking the native methods of the virtual machine (that is, those methods which have no bytecode representation). We will develop estimation techniques for this purpose.

Work carried out: Work on the use of Markovian analysis in estimating the impact of garbage-collection ignorant native methods in the JVM on managed object allocations has been completed and is now being written up for publication.

WP9: Reducing size of certificates; negotiation vs. proof

Objectives: Explore alternatives to 100%-guaranteed certificates when these are infeasible.

This package is concerned with exploring possible alternatives in situations where the generation and transmission of 100%-guaranteed certificates is unfeasible for one of the following reasons:

- certificates exist, but are prohibitively large

- certificates can in principle be obtained, but only at prohibitively high cost (time and human resource needed for theorem proving, runtime of program analyses)
- certificates can in principle not be obtained due to influence of unknown or merely estimable parameters.

This workpackage is more tentative and visionary than the other ones. Progress will be strongly dependent on the results obtained in the other packages and the particular qualifications of the research assistants.

A minimum deliverable will consist of visionary articles fleshing out the ideas thus enabling future interaction and perhaps collaboration with others working on these issues.

b/c. **Advanced techniques of certification** *(2.5 months)*
Deliverables: visionary paper *Prerequisites: 6a,2a/b*

Here we plan to depart from the requirement of endowing mobile code with mathematical proofs of resource bounds.

The first idea is to consider interaction-based probabilistic certification. The general scenario will be as follows: A sends B a piece of code to be executed remotely. On the basis of the code and possibly random data B computes a challenge (in the style of "please give letters 3 and 7 of your online password") to which A must respond. B may then accept the code, reject it, or set another challenge.

The theory of *probabilistically checkable proofs* [AMS$^+$92, Aro94] shows how based on a concept of polynomially-sized and polynomial time verifiable certificate (which we can assume here) such a protocol can be devised so that challenges have logarithmic size, responses have constant size, and the probability that A can give a correct response to a challenge although no certificate exists is $<50\%$ so that k independent rounds lead to an error probability of $\leq 2^{-k}$.

In spite of some encouraging recent work [HS00] the overhead on the side of A remains considerable. We plan to investigate feasibility of this approach in our context and study possible relaxations, for instance allow for larger size responses. We emphasize that our concern is not to advance the theory of probabilistically checkable proofs, but exclusively to harness it for the purpose of certification of mobile code.

A different idea is to consider certification by negotiation in the absence of rigorous proofs. The above-described protocol still requires that the sender A actually possesses a proof that his program meets the required bounds. It only reduces the amount of information that is actually interchanged. For the case where proofs are too difficult or impossible to obtain, we propose to investigate more liberal negotiation processes like the following:

- B provides a range of input parameters, A sends certificate which works only for this range.
- B challenges specific parts of the program. E.g. "you've got a write command in line XXX. Please convince me that this is fine." A then responds with a proof with assumptions that B may challenge again or accept.
- To cope with resource usage depending on unpredictable extraneous factors such as interaction patterns in concurrent systems or number of cache misses, we propose to investigate the use of probabilistic models such as PEPA [GH94, Hil96]. A and B would agree upon a probabilistic model to be used; A would then carry out the modelling and

would provide B with verifiable results obtained in this way, for instance in the form of probability matrices or selected rows/columns thereof.

Work carried out: A PCP framework for certification of arbitrary type systems has been formulated and is being implemented.

This package is appreciably more risky than the previous ones as we move further away from well-understood terrain in programming language theory. However, the possible reward will be high, as size of certificates and overhead in their production forms the biggest foreseeable obstacle against wide practical use of resource certification. If it can be successfully overcome or at least the necessary foundations laid, we will have paved the way for practical resource certification in the context of global computing.

2.3 Comparison of planned activities and actual work

Even though only a few technical deliverables were actually completed during the part of year 3 covered by this report, major progress has been made on all remaining tasks. For most of those planned for delivery during year 3, most work is complete and delivery is expected during the first two months of the extension period.

Referring to the specific objectives for year 3 listed in Section 2.1:

- "Complete work on heap space type inference for Camelot (task 7d)": **close to completion**
- "Extend work on the bytecode logic (deliverable D2a/b) to yield a component that will generate heap space bound certificates from space types (task 6a)": **achieved**

"Additional priorities during year 3 were other tasks from WP7 and scheduled tasks from WP8 and WP9": **4e/f achieved, 7e, 8a and 8d close to completion, 7c and 9b/c begun**

The following charts record the per-workpackage per-site per-annum activity. Manpower is given in person-months and refers only to the researchers paid by the project. The estimated manpower figures are incomplete, since the figures that were required for the Technical Annex were per-workpackage per-site totals, and per-task overall totals, but not the detailed per-workpackage per-site per-annum totals that would be most relevant here. Where it is possible to extract information from the figures in the Technical Annex (for instance, when a Workpackage is scheduled for activity in a single year) then this is recorded in the chart; otherwise the entry is left blank. In such cases, partial information may be obtained by comparing the cumulative totals with the planned totals over the entire project duration.

Comparison with the chart from last year's report show discrepancies in the figures for planned effort. This is due to the fact that the revised workplan adjusted the plan to fit what had actually happened in the project up to that point. As a consequence, the figures for "Planned" and "Actual" match exactly for workpackages that were completed before the revised workplan was submitted.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP1	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	12	12	0	0	0	0	12	12	12
LMUMUN	0	0	0	0	0	0	0	0	0
Total	12	12	0	0	0	0	12	12	12

	Year 1		Year 2		Year 3		Cumulative		Full project
WP2	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	2	2	16	16	0	0	18	18	18
LMUMUN	9.5	9.5	10	10	0	0	19.5	19.5	19.5
Total	11.5	11.5	26	26	0	0	37.5	37.5	37.5

	Year 1		Year 2		Year 3		Cumulative		Full project
WP3	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	3.5	3.5	8.8	8.8	0	0	12.3	12.3	12.3
LMUMUN	5.5	5.5	0	0	0	0	5.5	5.5	5.5
Total	9	9	8.8	8.8	0	0	17.8	17.8	17.8

	Year 1		Year 2		Year 3		Cumulative		Full project
WP4	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN		2		9		4	13	15	13
LMUMUN		4		2		5	11	11	11
Total		6		11		9	24	26	24

WP4 is now complete. The total amount of effort and the distribution of work between partners is approximately according to (the revised) plan.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP5	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	0	0	0	0	0	0	0	0	0
LMUMUN	1	1	3	3	0	0	4	4	4
Total	1	1	3	3	0	0	4	4	4

WP5 is now complete, since all remaining tasks have been eliminated from the revised workplan. Both partners were involved in this work but the only paid researchers involved were from the Munich site.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP6	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN		0		1.5		22.41		23.91	22.2
LMUMUN		0		9		15		24	24
Total		0		10.5		37.41		47.91	46.2

Total resource spent on WP6 so far is according to plan, but task 6g remains to be completed.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP7	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN		0		5		17		22	17
LMUMUN		0		0		0		0	5
Total		0		5		17		22	22

WP7 is roughly on track; several deliverables remain but most of the work is complete on the majority of them. Much of the progress that has been made here is the work of a PhD student in Munich (Steffen Jost) who is not employed by MRG.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP8	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN		0		0		8.46		8.46	12
LMUMUN		1.5		1		4		6.5	6.5
Total		1.5		1		12.46		14.96	18.5

WP8 is roughly on track.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP9	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	0	0	0	0	2.5	1.5	2.5	1.5	2.5
LMUMUN	0	0	0	0	0	0	0	0	0
Total	0	0	0	0	2.5	1.5	2.5	1.5	2.5

WP9 now contains only one task, which is on track.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP10	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	0	0	3.3	3.3	0	0	3.3	3.3	3.3
LMUMUN	0	0	0	0	0	0	0	0	0
Total	0	0	3.3	3.3	0	0	3.3	3.3	3.3

WP10 is complete, since the remaining task has been eliminated from the revised workplan.

	Year 1		Year 2		Year 3		Cumulative		Full project
WP11	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	0.5	0.5	0	0	0	0	0.5	0.5	0.5
LMUMUN	0	0	0	0	0	0	0	0	0
Total	0.5	0.5	0	0	0	0	0.5	0.5	0.5

WP11 is proceeding according to schedule. Most of the tasks are allocated 0 person-months because the work is being done by the main investigators rather than the researchers who are employed by the project.

	Year 1		Year 2		Year 3		Cumulative		Full project
ALL WP	Planned	Actual	Planned	Actual	Planned	Actual	Planned	Actual	Planned
UEDIN	30	20	30	43.6	30	53.37	90	116.97	90
LMUMUN	24	21.5	24	25	24	24	72	70.5	72
Total	54	41.5	54	68.6	54	77.37	162	187.47	162

When the workplan was revised, the budget and the allocation of money remained unchanged. Thus the *total* planned resource per site per year for the project remains as it was in the *original* workplan, despite the fact that the planned resource per year for most workpackages changed significantly in the revised workplan. The difference between planned man-months and actual man-months, without a change in the budget, is accounted for mainly by the appointment of more staff at the Edinburgh site that are more junior than anticipated in the original budget. In particular, a 50% post for a senior researcher at professorial level was included in the budget and was planned to be filled by the Project Coordinator who would be released from 50% of his normal duties to devote

this time to technical work on the project, but this turned out to be impossible under Commission rules. This increase in manpower was an important factor in getting the project back on schedule during year 2 when some tasks turned out to be much more difficult than anticipated, and it has also been important in keeping the project on schedule during year 3.

2.4 World-wide 'state-of-the-art' update

This section provides a brief account of particularly relevant technical developments world-wide, which do not already appear in the workplan or previous periodic progress report. We outline each of these developments, giving an evaluation and analysis of their impact on the project.

- The *Hume* project (`www.hume-lang.org`) continues to work on a high-level functional language for embedded systems. The Hume language is based on concurrent automata controlled by transitions characterised by pattern matching on inputs and (recursive) function generation on outputs. Recent work includes an automatic cost analysis based on a notion of sized types [VH04]. The bounds obtained are in terms of general recurrence relations which must be solved using a tool such as Mathematica. This approach is less amenable to automation than the MRG approach, but whereas the MRG methods can only provide linear bounds, the Hume method can produce estimates of heap usage which are non-linear in the size of the input.

- Chang et al [CCNS05] describe a extensible framework for verification of Proof Carrying Code. They present a system called the *Open Verifier* which consists of a trusted extensible verifier which allows the installation of (untrusted) extensions which can then be used to verify properties of proof-carrying programs. This is a very interesting initiative which goes a long way towards simplifying some of the engineering issues associated with the creation of PCC systems. The MRG project opted to use Isabelle proof scripts as certificates, and used the Isabelle theorem prover as the verifier on the consumer side of the PCC system. Our justification for this was that it sufficed as a proof of concept while avoiding time-consuming but essentially uninteresting engineering problems. The Open Verifier provides a means of producing verifiers which is very flexible and requires considerably less effort than the implementation of a custom verifier for a particular PCC system, and (had it been available at the time) would have provided an excellent alternative to the strategy used in MRG.

- More new work has been carried out on Reynolds' Separation Logic [Rey02]. In [BCOP05] Separation Logic is used to study race-free sharing of heap storage between concurrent threads. In [CGZ05] a logic called *Context Logic* is introduced which is shown to be a generalisation of Separation Logic; in contrast to Separation Logic, which is used to reason about heap update, Context Logic is used to reason about dynamic update of complex data structures such as the trees-with-pointers which occur in XML. Both of these papers deal with questions of resources which are broadly similar to the questions which are dealt with in MRG, but as yet we have not explored the connection in detail.

- At LMU München, Steffen Jost is in the process of extending the Hofmann-Jost heap-space usage inference algorithm [HJ03a] to include higher-order functions and a better treatment of resource bounds for polymorphic functions [Jos]. This is a significant extension: to date, most analyses of heap usage (in particular, those used by the MRG group) have only been able to deal with first-order functions. This restriction arises because higher-order functions

are usually implemented via closures which require heap memory to be allocated in some compiler-dependent and non-transparent manner. Another approach to this problem has been implemented within MRG, where the high-level Camelot language has been extended to include higher order functions via Reynold's *defunctionalisation* technique. This is a technique which transforms higher-order functional program into first-order ones by replaces closures by members of algebraic datatypes, and has the advantage that once a program has been converted to a first-order form it is amenable to the safety and resource usage analyses which we have already developed and formalised.

2.5 Previous review reports

In their report on the second year of MRG the reviewers made several comments. We note here the action we have taken in response.

The project structure should be rationalized as planned: This has been done; the revised Technical Annex is dated 8 October 2004.

Expose technology to potential users: This year, the main progress towards engaging potential users has been via projects funding collaborative work, see Section 5.

2.6 Planned work for the next reporting period

The next period of work consists of the final four-month extension to the project which was agreed so that the project would still be running at the time of the planned final review.

Our plan for the extension period is to complete all remaining tasks. These include tasks that were scheduled during year 3 but not completed during that period, and tasks that were scheduled for completion during the extension period. Specifically:

D6g: Total correctness, due April 2005

D7a/d: Extension of type system by allowing user annotations; methods to infer type annotations automatically, due June 2004

D7c: Extension of basic type system to object-oriented language, due September 2004

D7e: Extension of basic type system to mutable state and concurrency, due September 2004

D8a: Relationship between proof-based certificates and present-day security management, due October 2004

D8b: Certificate-based resource manager, due April 2005

D8d: Methods for estimating cost of native methods, due August 2004

D8e: Expressing and relating resource policies, due April 2005

D9b/c: Advanced techniques of certification, due December 2004

3 Project management and co-ordination

Cooperation within the consortium has been excellent. The fact that there are only two partners has meant that collaboration is strong.

Project meetings. MRG organized one internal project workshop during year 3. This was held in Melrose, near Edinburgh, during 11–13 October 2004 and was attended by Kevin Hammond of St Andrews University, who spoke about his work on the Hume project, in addition to project personnel. An open workshop involving all projects in the Global Computing pro-active initiative was held in Rovereto during 9–12 March 2004.

Collaboration. Most tasks involve contributions from both sites. Most collaboration is via e-mail but intensive periods of joint work have taken place during visits by reseachers at each site to the other site, as follows:

- Loidl to Edinburgh, 1–10 May 2004
- MacKenzie to Munich, 13–18 March 2004
- Beringer to Munich, 6–10 January 2004 and 7–14 February 2004

In addition, in most cases researchers travelling to project workshops spent extra days before or after the workshop at the host site for collaboration.

Personnel changes. The following personnel changes have taken place during year 3:

Bartek Klin spent 1.5 months in summer 2004 working on MRG at the Edinburgh site. During this period he made a major contribution to task 9b/c.

Lennart Beringer took a break of 3 months from MRG during 2004 to work on a different project in LFCS at Edinburgh that required his expertise.

Ulrich Schoepp joined the Edinburgh site in October; he is working 57% for MRG while studying for a PhD.

Cooperation with other projects.

- Martin Hofmann is coordinator of the APPSEM-II thematic network. APPSEM theme H is "Resource models and web data", which includes MRG, making APPSEM a useful forum for discussion of MRG and related work with other experts.
- There are several points of contact between MRG and the AGILE project. Martin Hofmann and Martin Wirsing (AGILE coordinator, Munich) are conducting preliminary investigations on an extension of the MRG framework to UML using AGILE technology. Don Sannella and Andrzej Tarlecki (AGILE, Warsaw) are collaborating on studying the application of work from AGILE in the MRG framework, with support from a travel grant funded by the British Council and the Polish Ministry of Scientific Research and Information Technology. David Aspinall is involved in a related investigation with Piotr Hoffman (AGILE, Warsaw).
- Don Sannella is a member of the MIKADO advisory committee.
- Roberto Amadio, who spent 6 months in 2002/2003 working on MRG at the Munich site, is a member of PROFUNDIS. We remain in contact with him since his return to Marseille and he has continued MRG-related research.

- Stephen Gilmore is a member of the DEGAS project. A case study from the DEGAS project (a multi-player on-line role playing game that runs on a Java-enabled mobile telephone) has been used in two MRG-related student project at Edinburgh.
- Hans-Wolfgang Loidl and Martin Hofmann are members of the consortium for EmBounded, an FET-Open STREP project on resource bounds in embedded systems that will start in early 2005.
- MRG continues to have useful interactions with the ConCert project at Carnegie Mellon University.
- Members of MRG have again been invited to attend CASSIS (Construction and Analysis of Safe, Secure and Interoperable Smart Devices) in Marseille during March, an invitation-only workshop organized by the Everest group at INRIA Sophia Antipolis. This is a prime opportunity for interaction with the main academic and industrial researchers in smart cards, and the invitation amounts to recognition by members of that community that work in MRG is applicable there.
- Bartek Klin, who spent 1.5 months in 2004 working on MRG at the Edinburgh site, has now joined MyThS. We remain in contact and he continues to collaborate on task 9b/c.
- MRG members collaborated with personnel from most projects in the Global Computing initiative on FoG, an unsuccessful FET-GC2 Network of Excellence proposal.
- MRG members collaborated with personnel from AGILE, DART, PROFUNDIS, SECURE and many non-GC projects on MOBIUS, a successful FET-GC2 Integrated Project proposal.

4 Cost breakdown

At the time of writing, detailed financial data is unavailable.

5 Information dissemination and exploitation of results

Dissemination of the results of MRG has been via publications, research talks and presentations, courses, the project website, software, organization of workshops and joint dissemination activities, and involvement with other projects at national and international level.

Publications. The following publications by MRG members that are directly relevant to the topic of the MRG project have appeared or been accepted during 2004.

Invited papers: [AGH+05]

Refereed journals and conferences: [ABH+04], [Atk04], [BHMS04], [BHMS05], [Gil04] [GMW05], [MW04], [WM04],

Non-refereed journals and conferences: [Loi04], [ML04]

Other publications by MRG members that have appeared or been accepted during 2004 are: [ABHS04], [AGM+04], [AL04], [BCGH04a], [BCGH04b], [BGHN04], [BGT04], [BST04], [BTL04], [CGH04], [GHHlK04], [GHK04], [GHKR04], [GKP04], [HS04], [LL04], [LMO04], [LS05], [MST04], [MT04], [PS04], [RTL04], [SS04], [ST04], [Sta05], [ZTLM04].

Research talks and presentations. Talks to present the conference papers listed above, plus the following:

- David Aspinall gave a talk on "Specification of Datatypes-in-Memory" at WADT 2004 during ETAPS 2004 in Barcelona.
- David Aspinall gave a talk on "Logics for Certifying Resource Bounds" at the University of Wales in Swansea.
- Robert Atkey gave a talk on "A Calculus for Resource Relationships" at SPACE 2004 during POPL 2004 in Venice.
- Robert Atkey gave a talk on "A Categorical Semantics of Resource Separation" at the APPSEM meeting in Tallinn, Estonia.
- Lennart Beringer gave a talk on MRG at the Types 2004 workshop in Paris.
- Martin Hofmann gave an invited talk on "Program Logic and Derived Assertions" at ICALP 2004 in Turku.
- Martin Hofmann gave a talk on "Certificate Generation in MRG" at SPACE 2004 during POPL 2004 in Venice.
- Martin Hofmann gave a talk on "Mobile Resource Guarantees" at the University of Freiburg and Aachen University.
- Martin Hofmann gave a talk on "Termination Logic" at an IFIP WG2.8 Meeting in West Point.
- Hans-Wolfgang Loidl gave a talk on "A Proof-Carrying-Code Infrastructure for Resource Guarantees" at Heriot-Watt University in Edinburgh.
- Hans-Wolfgang Loidl gave a talk on "Proving Bounded Resource Consumption for Mobile Code" at Johannes Kepler University in Linz.
- Alberto Momigliano gave a talk on "Certificate Generation in MRG" at the University of Milan.
- Don Sannella gave talks on "Mobile Resource Guarantees" at the University of Strathclyde, the University of Birmingham, Universitá Ca' Foscari in Venice, Australian National University in Canberra, Melbourne University, and the University of New South Wales.

Project website. The project website is at `http://groups.inf.ed.ac.uk/mrg/`.

Software. Grail and Camelot are available for download from the project website, and all other software produced by MRG will also be made available there.

Organization of workshops.

- Hans-Wolfgang Loidl was Programme Chair and Local Organiser of the Fifth Symposium on Trends in Functional Programming (TFP2004) in Munich.
- Martin Hofmann served on the Programme Committee of POPL 2005, ICALP 2004 and LICS 2004.
- Don Sannella was Programme Chair for ICALP 2004 in Turku and served on the Programme Committee of the 2nd ACM International Workshop on Mobility Management and Wireless Access Protocols (MobiWac).

Involvement with other projects. See Section 3 above.

Exploitation of results. A number of companies have shown interest in the results of the MRG project. These include: Gemplus, NTT and Trusted Logic (smart cards), BT Engineering (mobile telephones), Sun Labs (real-time Java), Helixion (secure multimedia), Motorola (automotive networks) and AbsInt (embedded systems), among others. In the closing stages of the project, we have been attempting to form concrete collaborative arrangements with some of these in the form of follow-on projects. So far, the following attempts to secure funding have been successful:

- EmBounded is an FET-Open STREP project on resource bounds in embedded systems that will start in early 2005, building on the results of the MRG project and specifically work on Grail and the Grail bytecode logic. Partners include LMU München and AbsInt.
- MOBIUS is a FET-GC2 Integrated Project proposal that will build on the results of MRG. Partners include Edinburgh and LMU München and industrial partners include Trusted Logic, France Telecom and SAP. An industrial User Panel includes about a dozen companies from a range of relevant sectors of industry.

An additional project proposal at Edinburgh on applying MRG techniques to the Grid is under evaluation at EPSRC, and discussions are in progress with Helixion concerning a possible project with Edinburgh on digital rights management.

References

[ABH+04] David Aspinall, Lennart Beringer, Martin Hofmann, Hans-Wolfgang Loidl, and Alberto Momigliano. A program logic for resource verification. In *Proceedings of 17th International Conference on Theorem Proving in Higher Order Logics (TPHOLs2004)*, volume 3223 of *Lecture Notes in Computer Science*, pages 34–49, Heidelberg, September 2004. Springer-Verlag LNCS.

[ABHS04] Klaus Aehlig, Ulrich Berger, Martin Hofmann, and Helmut Schwichtenberg. An arithmetic for non-size-increasing polynomial-time computation. *Theoretical Computer Science*, 318, 2004.

[AGH+05] David Aspinall, Stephen Gilmore, Martin Hofmann, Donald Sannella, and Ian Stark. Mobile resource guarantees for smart devices. In *Construction and Analysis of Safe, Secure, and Interoperable Smart Devices: Proceedings of the International Workshop CASSIS 2004*, number 3362 in Lecture Notes in Computer Science, pages 1–26. Springer-Verlag, 2005.

[AGM+04] Samson Abramsky, Dan Ghica, Andrzej Murawski, Luke Ong, and Ian Stark. Nominal games and full abstraction for the nu-calculus. In *Proceedings of the Nineteenth Annual IEEE Symposium on Logic in Computer Science*, pages 150–159. IEEE Computer Society Press, 2004.

[AL04] David Aspinall and Christoph Lüth. Proof General meets IsaWin — combining text-based and graphical user interfaces. In *International Workshop on User Interfaces for Theorem Provers (UITP'03)*, volume 104 of *Electronic Notes in Theoretical Computer Science*, 2004.

[AMS+92] S. Arora, R. Motwani, M. Safra, M. Sudan, and M. Szegedy. Proof verification and intractabilitity of approximation problems. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 13–22, 1992.

[Aro94] Sanjeev Arora. *Probabilistic Checking of Proofs and Hardness of Approximation Problems.* PhD thesis, UC Berkeley, 1994. UCB Technical Report: CS-TR-476-94.

[Atk04] Robert Atkey. A λ-calculus for resource separation. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004*, volume 3142 of *Lecture Notes in Computer Science*, pages 158–170. Springer, July 2004.

[BCGH04a] Anne Benoit, Murray Cole, Stephen Gilmore, and Jane Hillston. Evaluating the performance of pipeline-structured parallel programs with skeletons and process algebra. *Parallel and Distributed Computing Practices*, pages 1–19, 2004. Accepted for publication.

[BCGH04b] Anne Benoit, Murray Cole, Stephen Gilmore, and Jane Hillston. Evaluating the performance of skeleton-based high-level parallel programs. In *Proceedings of the 4th International Conference on Computational Science*, number 3038 in LNCS, pages 289–296, Kraków, Poland, June 2004. Springer-Verlag. Refereed conference paper.

[BCOP05] R. Bornat, C. Calcagno, P O'Hearn, and M Parkinson. Permission accounting in separation logic. In *POPL 2005: Conference Record of the 32nd Annual ACM Symposium on Principles of Programming Languages.* ACM Press, 2005.

[BCP97] Kim B. Bruce, Luca Cardelli, and Benjamin C. Pierce. Comparing object encodings. In *Theoretical Aspects of Computer Software (TACS), Sendai, Japan*, September 1997. An earlier version was presented as an invited lecture at the Third International Workshop on Foundations of Object Oriented Languages (FOOL 3), July 1996.

[BGHN04] M. Buchholtz, S. Gilmore, J. Hillston, and F. Nielson. Securing statically-verified communications protocols against timing attacks. In J. Bradley and W. Knottenbelt, editors, *Proceedings of the First International Workshop on Practical Applications of Stochastic Modelling*, pages 61–80, London, England, September 2004. Refereed workshop. Paper to appear in ENTCS.

[BGT04] Jeremy T. Bradley, Stephen T. Gilmore, and Nigel Thomas. How synchronisation strategy approximation in PEPA implementations affects passage time performance results. In M. Núñez *et al*, editor, *Applying Formal Methods: Testing, Performance, and M/E-Commerce (EPEW 2004)*, volume 3236 of *LNCS*, pages 128–142. Springer-Verlag, October 2004. Refereed workshop paper.

[BHMS04] Lennart Beringer, Martin Hofmann, Alberto Momigliano, and Olha Shkaravska. Towards certificate generation for linear heap consumption. In *Proceedings of ICALP/LICS Workshop on Logics for Resources, Processes, and Programs (LRPP2004)*, July 2004. To appear.

[BHMS05] Lennart Beringer, Martin Hofmann, Alberto Momigliano, and Olha Shkaravska. Automatic certification of heap consumption. In Andrei Voronkov Franz Baader, editor, *Logic for Programming, Artificial Intelligence, and Reasoning: 11th International Conference, LPAR 2004, Montevideo, Uruguay, March 14-18, 2005. Proceedings*, volume 3425 of *Lecture Notes in Computer Science*, pages 347–362. Publisher: Springer-Verlag GmbH, Feb 2005.

[BST04] Michel Bidoit, Donald Sannella, and Andrzej Tarlecki. Toward component-oriented formal software development: an algebraic approach. In *Proc. 9th Monterey Workshop, Radical Innovations of Software and Systems Engineering in the Future*, volume 2941 of *Lecture Notes in Computer Science*, pages 75–90. Springer, 2004.

[BTL04] A. Rauber Du Bois, P. Trinder, and H-W. Loidl. mHaskell: Mobile computation in a purely functional language. In *International Workshop on Implementation and Application of Functional Languages (IFL 2004)*, 2004.

[CCNS05] Bor-Yuh Evan Chang, Adam Chlipala, George C. Necula, and Robert R. Schneck. The open verifier framework for foundational verifiers. In *TLDI '05: Proceedings of the 2005 ACM SIGPLAN international workshop on Types in languages design and implementation*, pages 1–12. ACM Press, 2005.

[CGH04] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signaling pathway using the stochastic process algebra PEPA. In *Proceedings of BioConcur'04*, London, England, August 2004. Refereed workshop paper. Extended version submitted to Transactions on Computational Systems Biology and currently under review.

[CGZ05] C. Calcagno, P. Gardner, and U. Zarfati. Context logic and tree update. In *POPL 2005: Conference Record of the 32nd Annual ACM Symposium on Principles of Programming Languages*. ACM Press, 2005.

[CW00] K. Crary and S. Weirich. Resource bound certification. In *Proc. 27th Symp. Principles of Prog. Lang. (POPL)*, pages 184–198. ACM, 2000.

[GH94] Stephen Gilmore and Jane Hillston. The PEPA workbench: A tool to support a process algebra-based approach to performance modelling. In *Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Springer LNCS vol. 794*, pages 353–368, 1994.

[GHHlK04] Stephen Gilmore, Valentin Haenel, Jane Hillston, and Leïla Kloul. PEPA nets in practice: Modelling a decentralised peer-to-peer emergency medial application. In M. Núñez *et al*, editor, *Applying Formal Methods: Testing, Performance, and M/E-Commerce (EPEW 2004)*, volume 3236 of *LNCS*, pages 262–277. Springer-Verlag, October 2004. Refereed workshop paper.

[GHK04] S. Gilmore, J. Hillston, and L. Kloul. PEPA nets. In M.C. Calzarossa and E. Gelenbe, editors, *Performance Tools and Applications to Networked Systems: Revised Tutorial Lectures*, number 2965 in Lecture Notes in Computer Science, pages 311–335. Springer-Verlag, 2004. Extended version of an unrefereed tutorial paper.

[GHKR04] S. Gilmore, J. Hillston, L. Kloul, and M. Ribaudo. Software performance modelling using PEPA nets. In *Proceedings of the Fourth International Workshop on Software and Performance*, pages 13–24, Redwood Shores, California, USA, January 2004. ACM Press. Refereed conference paper.

[Gil00] Stephen Gilmore. Deep type inference for mobile functions. In P. Trinder G. Michaelson and H.-W. Loidl, editors, *Trends in Functional Programming (Volume 1)*, pages 40–48, 2000.

[Gil04] Stephen Gilmore. Extending Camelot with mutable state and concurrency. In *Proceedings of the 4th International Conference on Computational Science*, number 3038 in LNCS, pages 306–313, Kraków, Poland, June 2004. Springer-Verlag. Refereed conference paper.

[GKP04] S. Gilmore, L. Kloul, and D. Piazza. Modelling role-playing games using PEPA nets. In *Proceedings of the 19th International Symposium on Computer and Information Sciences (ISCIS 2004)*, volume 3280 of *LNCS*, pages 523–532, Kemer-Antalya, Turkey, October 2004. Springer-Verlag. Refereed workshop paper.

[GMW05] S. Gilmore, K. MacKenzie, and N. Wolverson. Extending resource-bounded functional programming languages with mutable state and concurrency. *Parallel and Distributed Computing Practices*, 2005. To appear. An earlier version of this paper appeared in the proceedings of ICCS 2004.

[Hil96] Jane Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[HJ03a] M. Hofmann and S. Jost. Static prediction of heap space usage for first-order functional programs. In *Proceedings of the 30th ACM Symposium on Principles of Programming Languages*, volume 38 of *ACM SIGPLAN Notices*, pages 185–197, New York, January 2003. ACM Press.

[HJ03b] Martin Hofmann and Steffen Jost. Static prediction of heap space usage for first-order functional programs. In *Proceedings of the 30th ACM Symposium on Principles of Programming Languages*, New Orleans, 2003.

[Hof00a] Martin Hofmann. Linear types and non size-increasing polynomial time computation. To appear in *Theoretical Computer Science*. See `www.dcs.ed.ac.uk/home/papers/icc.ps.gz` for a draft. An extended abstract has appeared under the same title in Proc. Symp. Logic in Comp. Sci. (LICS) 1999, Trento, 2000.

[Hof00b] Martin Hofmann. A type system for bounded space and functional in-place update. *Nordic Journal of Computing*, 7(4):258–289, 2000.

[HP99] J. Hughes and L. Pareto. Recursion and dynamic data structures in bounded space: towards embedded ML programming. In *Proc. International Conference on Functional Programming (ACM). Paris, September '99.*, pages 70–81, 1999.

[HS00] Prahladh Harsha and Madhu Sudan. Small pcps with low query complexity. *Electronic Colloquium on Computational Complexity*, 2000. Report No. 61.

[HS04] Martin Hofmann and Phil Scott. Realizability models for bll-like languages. *Theoretical Computer Science*, 318:121–137, 2004.

[Jos] Steffen Jost. Arthur: A resource-aware typesystem for heap-space usage reasoning. See `http://www.tcs.informatik.uni-muenchen.de/~jost/publication.html`.

[Kon03] Michal Konečný. Functional in-place update with layered datatype sharing. In *Proceedings of the 6th International Conference on Typed Lambda Calculus and Applications*, number 2701 in Lecture Notes in Computer Science, pages 195–210, Valencia, 2003. Springer-Verlag.

[LL04] M. Lange and H-W. Loidl. Parallel and Symbolic Model Checking for Fixpoint Logic with Chop. In *PDMC'04: Intl. Workshop on Parallel and Distributed Techniques in Verification*, PDMC'04: Intl. Workshop on Parallel and Distributed Techniques in Verification, London, UK, September 2004. To appear in ENTCS.

[LMO04] Kung-Kiu Lau, Alberto Momigliano, and Mario Ornaghi. Constructive specifications for compositional units. In *LOPSTR 2004 pre-proceedings*, 2004.

[Loi04] H-W. Loidl. A Resource-aware Program Logic for a JVM-like Language. In *Kolloquium Programmiersprachen und Grundlagen der Programmierung 2004*, volume Technical Report 213 of the Institute for Informatics, Univ of Freiburg, Germany, 17-19 March, 2004.

[LS05] Sam Lindley and Ian Stark. Reducibility and $\top\top$-lifting for computation types. In *Typed Lambda Calculi and Applications: Proceedings of the Third International Conference TLCA 2005*, Lecture Notes in Computer Science. Springer-Verlag, 2005. To appear.

[Mil78] Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348–375, August 1978.

[ML04] K. MacKenzie and H-W. Loidl. *A Gentle Introduction to Camelot.* LFCS, Univ of Edinburgh & Inst f Informatics, LMU Univ Munich, September 2004. Draft.

[MST04] Till Mossakowski, Donald Sannella, and Andrzej Tarlecki. A simple refinement language for CASL. In *Recent Trends in Algebraic Development Techniques: Selected Papers from WADT 2004*, volume 3423 of *LNCS*, pages 162–185. Springer, 2004.

[MT04] Alberto Momigliano and Alwen Tiu. Induction and co-induction in sequent calculus. In *Types for Proofs and Programs*, Lecture Notes in Computer Science. Springer-Verlag, 2004.

[MW04] Kenneth MacKenzie and Nicholas Wolverson. Camelot and Grail: resource-aware functional programming on the jvm. In *Trends in Functional Programing*, volume 4, pages 29–46. Intellect, 2004.

[PS91] Jens Palsberg and Michael Schwartzbach. Object-oriented type inference. In *Proc. ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, pages 246–161, 1991.

[PS04] John Power and Olha Shkaravska. From comodels to coalgebras: State and arrays. In *CMCS'2004: 7th Intl. Workshop on Coalgebraic Methods in Computer Science 2004*, volume 106 of *Electronic Notes in Theoretical Computer Science*, 2004.

[PWO97] Jens Palsberg, Mitchell Wand, and Patrick O'Keefe. Type inference with non-structural subtyping. *Formal Aspects of Computing*, 9:49–67, 1997.

[Rey78] J. C. Reynolds. Syntactic control of interference. In *Proc. Fifth ACM Symp. on Princ. of Prog. Lang. (POPL)*, 1978.

[Rey02] John Reynolds. Separation logic: A logic for shared mutable data structures. In *LICS 2002: Proceedings of the Seventeenth Annual IEEE Symposium on Logic in Computer Science*, pages 55–74, 2002.

[RTL04] A. Rauber Du Bois, P. Trinder, and H-W. Loidl. Towards Mobility Skeletons. In *CMPP'04 — Intl. Workshop on Constructive Methods for Parallel Programming*, Stirling, Scotland, July 2004. To appear in PPL.

[SS04] Ulrich Schöpp and Ian Stark. A dependent type theory with names and binding. In *Computer Science Logic: Proceedings of the 18th International Workshop CSL 2004*, number 3210 in Lecture Notes in Computer Science, pages 235–249. Springer-Verlag, 2004.

[ST04] Donald Sannella and Andrzej Tarlecki, editors. CASL semantics. In Peter Mosses, editor, *CASL Reference Manual*, volume 2960 of *Lecture Notes in Computer Science*, pages 115–274. Springer, 2004.

[Sta05] Ian Stark. Free-algebra models for the π-calculus. In *Foundations of Software Science and Computation Structures: Proceedings of FOSSACS 2005*, number 3441 in Lecture Notes in Computer Science, pages 155–169, 2005. To appear.

[TJ92] Jean-Pierre Talpin and Pierre Jouvelot. The type and effect discipline. In *Seventh Annual IEEE Symposium on Logic in Computer Science, Santa Cruz, California*, pages 162–173, Los Alamitos, California, 1992. IEEE Computer Society Press.

[VH04] Pedro Vasconcelos and Kevin Hammond. Inferring costs for recursive, polymorphic and higher-order functional programs. In *IFL 2003: Proceedings of the 15th International Workshop on the Implementation of Functional Languages*, Lecture Notes in Computer Science. Springer-Verlag, 2004. To appear.

[Wad90] Philip Wadler. Linear types can change the world. In *TC 2 Working Conference on Programming Concepts and Methods (Preprint)*, pages 546–566, 1990.

[WM04] Nicholas Wolverson and Kenneth MacKenzie. O'Camelot: Adding objects to a resource aware functional language. In *Trends in Functional Programing*, volume 4, pages 47–62. Intellect, 2004.

[ZTLM04] A. Al Zain, P.W. Trinder, H-W. Loidl, and G.J. Michaelson. Grid-GUM: Towards Grid-enabled Haskell. In *International Workshop on Implementation and Application of Functional Languages (IFL 2004)*, 2004.