CerCo

INFORMATION AND COMMUNICATION
TECHNOLOGIES
(ICT)
PROGRAMME

Project FP7-ICT-2009-C-243881 CerCo

Report n. D6.2
Plan for the Use and dissemination of foreground

Version 1.0

Main Authors:
C. Sacerdoti Coen

Project Acronym: CerCo
Project full title: Certified Complexity
Proposal/Contract no.: FP7-ICT-2009-C-243881 CerCo

Contents

1 Premise

The deliverable D6.2 *Plan for the Use and dissemination of foreground* is the first **report** of the CerCo project. The deliverable is **confidential** and required an effort of about **2 person-months**. The following is a description of the deliverable in the Grant Agreement:

> Plan for the Use and dissemination of foreground: An articulated dissemination plan, containing the individuation of the main target communities, and the relative exploitation strategies.

2 Overview and General Approach

2.1 Overview of expected results

Driven by a growing confidence in formal methods in general and interactive theorem proving in particular, critical IT systems are slowly undergoing a paradigm shift: testing of critical properties is going to be partially replaced with formal verification of programs written in high level languages, and a set of formally verified tools (interpreters, compilers, operating systems, communication libraries) is going to be used to preserve the verified properties for the execution of the low level code.

The present state of the art already provides a few mildly optimizing verified compilers that compete with traditional highly optimised compilers, and a few prototypes and experiments in the verification of operating systems. However, the only properties that have been formalized are extensional, proving that the high level semantics of the program—usually specified in

terms of infinite resources (e.g. memory)—is respected. All intensional properties that refer to resource consumption are not currently preserved during compilation or, at least, no formal proof of preservation is provided. Hence high level programs still need to be extensively tested to verify resource related properties like memory consumption or reaction time for reactive and real time programs.

In CerCo we will address intentional properties of programs, in particular execution time, and we will develop the first certified compiler that is able to manifest in the source program some intentional properties, mainly execution cost, that are inferred from the target program. The main challenge is to preserve the compositionality of compilation: whilst the source program undergoes several intentionally disruptive translations toward lower and lower level intermediate languages, we must keep track of the evolution of sensible code fragments in order to relate source, intermediate and target code fragments, so that execution costs may be back-propagated. Hence we expect to study new more refined ways of expressing the operational semantics of programs and to give a more refined notion of operational bisimulation, with the aim of obtaining a characterisation that is loose enough to capture some of the most common and effective optimisations.

Once we can manifest in the source program execution costs computed on the target program we will start investigating the exploitability of the cost annotations. In particular, cost annotations must be combined with extensional program invariants in order to obtain intensional program invariants, and be able to formally prove intensional properties of programs. We expect this study to require a long effort and to be performed outside the project timeframe. However, we assume that a combination of interactive theorem proving and automatated formal methods (like abstract interpretation) will need to be exploited and, in the project, we will focus on interactive theorem proving by exploiting and providing basic tools to reason about cost invariants.

All the major deliverables of CerCo are prototypes—either software prototypes or formal certifications—that are later integrated at project milestones into major CerCo compiler snapshots. These snapshots are the most appropriate deliverables for dissemination in the large, and different target communities are associated to the snapshots. We will discuss the milestones and snapshots in detail in Section 4 in order to identify the potential target communities and most appropriate means of dissemination.

Milestone name	Month due
Untrusted cost-annotating compiler	12
Untrusted CerCo compiler	24
Trusted CerCo compiler	36

2.2 Academic exploitation

Academic partners will take great advantage of the CerCo's results mainly in terms of increased technical know-how and scientific knowledge, increased visibility in the scientific community of reference, increased expertise, and for institutional academic exploitation (e.g. didactic activities). We also expect that the rest of the academic community will take up the innovative

ideas of the project, building on top of them code analyses at a level of accuracy that could have not been previously possible. Finally, we expect our methodology to be applied to other styles of high-level programming language (for instance, functional or logic) and to more complex scenarios involving, for example, a real time operating system.

2.3 Industrial exploitation

The long term industrial exploitation of the results of the CerCo project is mainly embodied in the area of embedded systems/software, in particular in the case of safety critical applications and time critical (realtime) applications. In the short term, software houses producing compilers for embedded systems could immediately draw benefit from the CerCo cost annotating technology or, more generally, by the knowledge gained in the certification of compilers. It is worth noting that several of these software houses are located in Europe, such as the medium-scale System Engineer Group of Freescale, headquartered in Scotland, Raisonance and Cosmic Software, headquartered in France, and the small-scale Hightech, headquartered in Germany, amongst many others.

2.4 Management of knowledge and intellectual property

All project partners signed a Consortium Agreement before the start of the project, legally confirming the principles of consortium management and mutually agreeing the precise relationship between the partners and their respective responsibilities for the duration of the work. In particular, the agreement includes specific arrangements with regard to intellectual property rights, to be applied amongst the participants and their affiliates, in compliance with the general arrangements stipulated in the contract. It thus specifies the rules for dissemination and use (confidentiality, ownership of results, patent rights, exploitation of results, and protection and dissemination of knowledge), as well as financial and legal provisions. A peculiarity of CerCo is that we are self-constrained to release open-source software only, which is also manifest in our prototypes that will all be made publicly accessible, nor did we identify any background to be protected.

2.5 Approach to Dissemination and Use

The CerCo project represents an unprecedented effort to create a verified chain of trust for intentional program properties during compilation, and we thus expect that, in the short term, it will generate a significant amount of interest in academia and, in the longer term, a significant amount of interest in industry, as well.

Dissemination and use of foreground will have a high priority that will be intensified towards the end of the project and whose target will progressively shift from academia to potential end users. Work package 6 will be devoted to all activities relevant to accomplishing this task.

We have planned appropriate measures to ensure an effective and timely dissemination of the project results to potential users, both at the European level and worldwide. Since the research started in CerCo is of the long term kind, the main target of dissemination will be research institutions that may immediately benefit from our results, and possibly take over the research effort after the termination of the CerCo project, to achieve long term results. We will also target industry by means of specific dissemination actions towards the end of the

project in order to advertise our results, and to ensure that the final outcome will fulfil the specific requirements of the industrial community.

Dissemination will be carried out by a variety of means:

- Talks at relevant international conferences, events and forums (both presenting the technical achievements and introducing at a high level the projects objectives and results).
- Publication of papers in proceedings of international conferences and events, as well as in international scientific journals.
- Organisation of meetings on project-related topics, including 'project workshops' where attendance of external experts and professionals is based on invitation.
- Organisation of at least one special event targeted at the scientific community, in the form of an open workshop, possibly affiliated to a major conference, or in the form of courses given at summer schools or equivalent teaching events targeted to PhD students.
- Organisation of at least one special event targeted at potential industrial stakeholders, in the form of a dissemination meeting to be hold during the last year of the project.
- Design and management of a publicly available website that includes descriptions of the main project results and allows the download of all the software and publications developed in CerCo. The web-site will be managed as a Wiki to maximise interaction with the community of users and researchers interested in the project.
- Press conferences and press releases describing the advancement and main results of the project, so as to reach and make the public aware of both the short-term and long-term impact of the project results.
- Dissemination of project objectives and results to members of European or national level projects having CerCo members as participants.

3 Description of the Dissemination Plan

3.1 Web presence and information exchange

The website of the CerCo project is

`cerco.cs.unibo.it`

The website combines a private and public Wiki. The Wiki will progressively include:

- A general introduction to the project: the objectives, the expected results, the milestones, the detailed description of the consortium and its position within the Seventh Framework Programme.
- The list of events taking place in the context of the project: meetings, conferences, workshops, and their availability to the public.
- Publications originating with the project, both in the scientific community and in the general press.

- A number of relevant links: other projects, institutions and companies that are related to CerCo.
- A public section containing the public deliverables, comprising the open source software developed in CerCo. In order to make a first trial of the software as simple as possible for interested users, we plan to provide packages for some common Linux distributions and a Live CD for non Linux users.
- A private section containing the remaining deliverables and other documents for the European Commission.
- A private section containing contact details, mailing lists archives, details about meetings held (slides, notes and so on) and other temporary technical information needed by the consortium.

Besides the website, communication and information exchange amongst the members of the project is enforced via a carefully organized and maintained central repository and a number of dynamically created mailing lists.

Ad hoc advertisement of CerCo to the interested research communities will also be achieved by advertisements sent to the mailing lists of the projects, as well as advertisement of open job positions.

3.2 Project Workshops and Conferences, Lectures, Tutorials

Two Project Workshops will be organized every year. They will consist of a public and a private chapter and we will invite as guests a small number or international researchers who are working on related problems. We will try to co-locate later workshops with international events in order to improve visibility of the project. During the third year we also plan to organise two separate additional events targeted at the scientific community and to potential industrial stakeholders.

Moreover, we aim to present our work at international conferences and forums on interactive and automatic theorem proving, resource aware computing, invariant generation, formal methods and reactive computing. In accordance with the character of the corresponding meeting, the presentation of CerCo may be part of a more comprehensive presentation, it may get a presentation of its own, or even only special aspects of CerCo may be addressed by a talk. In the latter case some member of the group with expertise in the respective task or workpackage may be the most suitable person to give the talk. Talks at larger conferences should be accompanied by papers on CerCo in the corresponding proceedings.

3.3 Publications

We aim to publish the results obtained in CerCo in the proceedings of international events and in international journals. The list of journals that publish papers related to the topics studied in CerCo comprises, but it is not limited to, the following ones:

- Journal of Automated Reasoning
- Journal of Formalized Reasoning
- Information and Computation

- Higher-Order and Symbolic Computation
- Communications of the ACM
- Science of computer programming

Many international conferences and workshops deal with compilation, formal reasonings and resource analysis. Some of the most important are certainly

- Principles of Programming Languages (POPL)
- Interactive Theorem Proving (ITP, ex TPHOLs)
- Logic for Programming, Artificial Intelligence and Reasoning (LPAR)
- Compiler Construction (CC)
- International Symposium on Formal Methods (FM)
- Symposium on Programming Languages and Reasoning (APLAS)

3.4 Clustering

This section indicates some projects and organisations relevant to the ongoing work (at worldwide, European or national level) and with which information exchange may be beneficial.

3.4.1 National/Local Projects and Research Groups

The following national/local projects and research groups are the most direct interlocutors for CerCo.

- CompCert `http://compcert.inria.fr`: the most advanced, closed source project on the development of a certified compiler for a subset of C, and a natural source of inspiration (and competitor) for CerCo.
- Frama-C `http://frama-c.com`: an extensible and collaborative platform dedicated to source-code analysis of C software. It is a natural candidate for a technological integration with the CerCo compiler, in order to provide high-level reasoning tools on the intentional properties of C programs.
- AbsInt `http://absint.com`: a private company located in Saarbruecken, Germany, that participated in several EU projects and has a good record of collaboration with the scientific community. It develops advanced development tools for embedded systems, and tools for validation, verification and certification of safety-critical software.

3.4.2 EU Projects

This is a selection of EU projects (mainly from FP7) dealing with embedded or, more generally, resource constrained systems.

- ARTEMIS: European Technology Platform/Joint Technology Initiative on Advanced Research & Technology for Embedded Intelligence and Systems

- Embounded: FP6 Strep on Automatic Analysis of Bounded Resources for Embedded Systems.
- ARTISTDesign: FP7 Network of Excellence on Design of Embedded Systems
- MNEMEE : FP7 Collaborative Project on Memory management technology for adaptive and efficient design of embedded systems. Jan 2008–Dec 2010.
- INTERESTED: FP7 Collaborative Project on Interoperable embedded systems Toolchain for enhanced rapid design, prototyping and code generation. Jan 2008–Dec 2010.
- PREDATOR : FP7 Collaborative Project on Design for predictability and efficiency. Feb 2008–Jan 2011.
- QUASIMODO : FP7 Collaborative Project on Quantitative system properties in model-driven design of embedded systems. Jan 2008–Dec 2010.

4 Description of the Use Plan (by result)

In this section, we describe the expected project results that are released in correspondence with project milestones and that have potential for exploitation.

4.1 Untrusted cost-annotating compiler

Delivery date: January 2011
Description: The untrusted cost-annotating compiler will be a functional compiler from a large subset of C to some assembly language. It will already trace the evolution of selected high-level code fragments in order to relate them to low level code fragments, and it will manifest cost annotations in the source language.
Target communities: compiler developers; developers of code invariant generators; developers of tools for cache behaviour prediction and general analysis of assembly code.

Compiler developers will be interested in the techniques used to trace evolution of code fragments during compilation. They should also be interested in suggesting enhancements to the proposed technique, in order to address more optimisations.

Developers of code invariant generators will be able to apply and extend their own techniques to deal with exact execution time by considering cost annotations during the invariant generation phase.

Developers of tools for cache behaviour prediction and general analysis of assembly code should be particularly interested since our approach allows to reflect at the source code level their own analyses performed on low-level target code.
Dissemination: Information dissemination for this activity will be done at the general project level, but will also try to target specific communities by targeted presentations/participation at relative conferences or meetings. In particular we plan to invite selected members of those communities to the first annual CerCo meeting to advertise the project results and foster potential collaborations.

4.2 Untrusted CerCo compiler

Delivery date: January 2012
Description: The CerCo compiler will be a re-implementation of the untrusted cost-annotating compiler in the variant of the Calculus of (Co)Inductive Constructions implemented by Matita, ready to be certified. Moreover, it will enhance the previous prototype by integrating a new layer for the management of cost annotations (produced by the compiler) and the complexity assertions (added by the user or synthesized automatically by abstract interpretation algorithms) in order to produce the right complexity obligations, that is, the goal to be proved in order to check the correctness of the assertions.
Target communities: the interactive theorem proving community; developers of code invariant generators; the abstract interpretation community; end users.

The interactive theorem proving community will benefit from an executable specification of a realistic compiler for C written in the Calculus of (Co)Inductive Constructions. However, at this stage the main benefit for the community will be a project by-product, which is a low level executable semantics (an interpreter) for some target machine and for several intermediate languages, that could easily be reused in other formalizations and translated to other interactive theorem provers.

Developers of code invariant generators and the abstract interpretation community should be interested in the new layer of management of cost annotations and complexity assertions in order to interface their techniques on top of it or to propose alternative management approaches. The prototype could also be used as a test-bench generator for their techniques.

Starting with the release of the prototype we will begin to contact potential end users, i.e. developers of time critical software, in order to attract feedback and to ensure that the final outcome will fulfil the specific requirements of the targeted exploitation community.
Dissemination: Information dissemination for this activity will be done at the general project level, with particular attention to the interactive theorem proving community. We will continue the practice of inviting selected members of the interested communities to the annual CerCo meetings in order to advertise the project results and foster potential collaborations. Moreover we will gradually enlarge the scope of the dissemination activity from developers and researchers in formal methods to the wider audience of users of compilers that are potentially interested in exact execution time, even if at this stage we do not expect the prototype compiler to apply to realistic programs.

4.3 Trusted CerCo compiler

Delivery date: January 2013
Description: The trusted CerCo compiler is the final prototype of CerCo and it is composed of several components:

- a proof of correctness in Matita of the compliance of the compiler, described as preservation of both the extensional and intensional behaviour of the program;
- an extension of the interactive theorem prover Matita with *ad hoc* tactics and mechanisms targeted toward the progressive automation—either in the small or in the large—of complexity proofs;

- an extensive use case dedicated to the automatic generation of cost invariants for the C code generated by a synchronous language compiler. The functionalities developed in the use case will be tested to compute certified reaction time bounds for significant examples of synchronous programs.

Target communities: the interactive and automatic theorem proving community; developers of code invariant generators; end users, in particular synchronous languages programmers.

The interactive theorem proving community will benefit from the first fully open source proof of correctness of a realistic compiler for C written in the Calculus of (Co)Inductive Constructions, and for a new technique for formally verifying preservation of intentional properties without losing precision. Developers of both interactive and automatic theorem provers will be interested in testing and enhancing their techniques when applied to automation of complexity proofs.

The final prototype of the CerCo project is supposed to be the first realistic milestone towards the formal certification of exact execution time (or exact reaction time) for real time or synchronous programs and thus it is meant to attract interest from end users. In particular, we will be looking for feedback on the applicability of the proposed technique, and we will target in particular the sub-community of synchronous programmers.

Dissemination: We will increase our dissemination activity during the final year of the project, in particular organizing separate events targeted at the scientific community and to potential industrial stakeholders. We will also improve accessibility and visibility of the CerCo compiler by developing packages for Linux distributions, and Live CDs for non-Linux users.