



CerCo

INFORMATION AND COMMUNICATION
TECHNOLOGIES
(ICT)
PROGRAMME

Project FP7-ICT-2009-C-243881 CerCo

Report n. D6.1
Project Web Site and Software Repository

Version 1.0

Main Authors:
M Puech, C. Sacerdoti Coen

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

Contents

1	Premise	2
2	Overview	2
3	Project “Web Site” (Trac)	3
4	Software Repository	4
5	Continuous Building System	4
6	Future evolutions	5

1 Premise

The deliverable D6.1 *Project Web Site and Software Repository* is the first **prototype** of the CerCo project. The deliverable is **public**, although part of the data handled by it are private, and required an effort of about **3 person-months**. The description of the deliverable in the Grant Agreement is the following one:

A web site will be designed and realised, and each partner will contribute to its content. The site will comprise a public part, accessible to everyone, and a private one, reserved to the project partners for the exchange of documentation, material, and to support project related activities. We will also install a revision control system (like SVN or GIT) to keep track of software evolution and we will adapt an existent continuous building system to trace software regressions nightly, in batch mode.

2 Overview

The aim of D6.1 is twofold: it must provide the main technological infrastructure for the advertisement and dissemination of the research results of CerCo, and it must also provide the means for direct communication and resource sharing among partners. Moreover, UNIBO will provide to the other partners the Matita[1] interactive theorem prover that will be used to obtain the certified CerCo compiler. We expect to make Matita evolve during the project timeframe in order to fix bugs that are critical for CerCo and in order to implement ad-hoc features. It is fundamental to grant to our partners immediate access to the changes of Matita and to monitor the behaviour of the system on the CerCo sources to avoid regressions and bugs introduction. Therefore, we plan to achieve a strict integration between the technological infrastructure for the development of Matita and the one for CerCo.

In order to target the previous objectives, UNIBO (the project coordinator) has bought a server to be dedicated to the project with the following characteristics:

Dell PowerEdge R410, 16GB Memory for 1 CPU, DDR3, 1333MHz, Intel Xeon X5570, 2.93Ghz, 8M Cache, 4 x 300 GB, 15K RPM Hard Drive (HOT SWAP), RAID controller.

The server characteristics have been chosen to allow for multiple virtual machines. In particular, for safety and flexibility purposes, direct access to the server is not allowed, but the server currently exposes the following virtual machines that can be accessed from the outside:

cerco.cs.unibo.it This virtual machine hosts the “project web site” (actually, a Trac instance [3]) and the software repository. These services are discussed in Sections 3 and 4. We installed on it the *stable* version of the Debian distribution of Linux in order to maximise availability.

cbs.helm.cs.unibo.it This virtual machine hosts the continuous building system (CBS) for Matita and CerCo that is discussed in Section 5. We installed on it the *unstable* version of the Debian distribution in order to immediately exploit the most recent developments of the software library that will be used in CerCo.

maelstrom.helm.cs.unibo.it This is the development virtual machine of CerCo. It is open to all project participants in order to do remote development and testing in a controlled environment that has all the required libraries and tools. It is also based on the *unstable* version of the Debian distribution and it is configured with all the libraries required by Matita and the OCaml ones we plan to use for CerCo .

The Department of Computer Science of the University of Bologna also provides an archived mailing list dedicated to the project members for general discussions and coordination. The archives are private and accessible to project members only.

To maximise fault tolerance, we configured the server with RAID 5, so that the loss of one hard disk does not compromise the data. Moreover, we provide incremental nightly backups of all the sensible data and we store the backups on a different physical machine. The main motivation is to be able to recover accidentally deleted files or old machine configurations.

Finally, the CerCo server acts as a network file system, NIS and DNS server for a cluster of virtual and physical machines (at the moment all located in Bologna) that can also be used for CerCo development.

We discuss now the implementation of the “Web Site”, Software Repository and Continuous Building System.

3 Project “Web Site” (Trac)

In order to ease the creation of a community around the CerCo prototype, it was decided to prefer a Wiki-centred solution to a traditional Web Site. Our choice has been to install a personalised version of the Integrated SCM & Project Management tool Trac [3], accessible at <http://cerco.cs.unibo.it>. Trac provides in an integrated and lightweight solution:

1. A Wiki for the advertisement and dissemination of the project results, and for fostering discussions and contributions from the user community.
2. An issue tracking system that subsumes a bug tracking system and provides a basic control over the project advancement.
3. An on-line, simplified Web interface to the software repository.

4. A flexible system of plug-ins to add new functionalities or modify the behaviour of the basic ones.

By means of additional plug-ins, we have integrated the public part of the Wiki with a private one. The private part is meant to integrate the mailing list in facilitating the discussion between the project developers. Moreover, since it is integrated with the software repository and Issue tracking systems, it is possible to create hyper-links to or to embed in discussion pages both code fragments and description of issues.

4 Software Repository

After considering several alternatives, it was decided to use a centralised software repository for the CerCo project, and we installed Subversion [2] on the CerCo server. Project members that prefer a distributed solution or that are used to work without network connectivity are advised to use the svn-git software to mirror the centralised repository.

The repository is open in read mode to anonymous users, and in write mode to project members only. A simplified web interface to the repository is also integrated in Trac and publicly available.

5 Continuous Building System

In CerCo we will develop both certified code (in Matita) and standard one (mainly in OCaml). We plan to adopt traditional testing methods (e.g. unit testing) to control bug regressions in the standard code, while the certified code will be checked by asking Matita to verify all the definitions and proofs of correctness. The last operation in particular can require quite a lot of time (in the order of at most a few hours), especially if proof automation will be heavily used in Matita to reduce the proving effort. Moreover, as already explained, we expect to observe a continuous evolution of Matita (and its standard library) to parallel the development of the CerCo. Each change in the code of Matita must trigger not only the checking for regressions on the CerCo code, but also on all the existent developments made in Matita (in order to avoid the possibility of introducing a bug that jeopardise the whole project or that will later require other changes before the next official release of Matita).

Since re-checking all the Matita developments already requires several hours even on a fast machine, we cannot expect the developers of CerCo to delay distribution of their changes to the partners until all the regression tests have been performed. Moreover, it is fundamental to keep trace of the evolution of the performances of both the code of Matita and the code of the CerCo prototype. Performance checks are meaningful only if systematically repeated always on the same machine and with reproducible load conditions. Hence we put up an ad-hoc continuous building system for Matita+CerCo that supports the following workflow:

- Commits are always allowed and developers are invited to commit as soon as they obtain a code that compiles and seems to implement the wanted changes on ad-hoc tests. When the commit is supposed to close a bug, the developer is also requested to commit a test for the bug to avoid future regressions. Development branches are encouraged on personal copies of the repository (using git-svn), but are discouraged on the central repository.
- Every night the CBS virtual machine is brought back to a pristine condition and a script checks-out the latest version of the Matita and CerCo code, compiles the code of Matita,

verifies the standard library, all the contribs of Matita and the certified CerCo code, and run all the regression tests on both the certified and un-certified code. All faults are recorded, as well as the performance of Matita on each source file.

- The statistics collected every night are stored in a data-base (a mysql instance) that runs on the cerco.cs.unibo.it virtual machine. The Subversion revision number is associated to the results in the statistics. Another set of scripts compares the statistics with those collected the previous night and, in case of significant regressions, sends a notification to all developers (using the CerCo mailing list).
- The developers can consult an ad-hoc web-interface to the data-base (written in python) that implements a set of ad-hoc queries or that allows the user to submit his own (by writing SQL code). In particular, the following predefined queries are available among others:
 1. Showing the list of commits between two given dates. In particular, the user can discover who is responsible for a commit that introduced a regression or that had a negative impact on performances.
 2. Compare the statistics between two dates or revision numbers.
 3. Monitor the performance evolution of the system on a single test.

On purpose, the CBS does not retract the commits that introduce regressions. We expect that the developers that are responsible for regressions will retract their code the very next day, or they will fix the regression or they will notify the other developers on the reasons for not fixing the regression (e.g. when the regression is actually caused by an already present bug that was previously masked by some other fault).

6 Future evolutions

The software architecture implemented so far is fully sufficient to support the initial development of CerCo. Moreover, the public part of the Wiki has been loaded with an overall description of the project that was largely obtained by the grant agreement. We expect the system to keep evolving during the lifetime of CerCo both in the public (and private) contents and in the code of the continuous building system. We will also consider in the future the opportunity of integrating other services in the Trac interface, like live demos of the CerCo prototype.

References

- [1] The Interactive Theorem Prover Matita, <http://matita.cs.unibo.it>
- [2] The Version Control System Subversion, <http://subversion.apache.org>
- [3] The Integrated SCM & Project Management tool Trac, <http://trac.edgewall.org>