mancoosi
managing software complexity

Collaboration Activities and Updated Collaboration Plan

Deliverable D7.3

Nature : Deliverable

Due date : 31/01/2010

Delivery Date: 10/02/2010

Start date of project : 01/02/2008

Duration : 36 months

Distribution: Public

SEVENTH FRAMEWORK PROGRAMME

Information Society Technologies

Web site: www.mancoosi.org

Blog: blog.mancoosi.org

Specific Targeted Research Project
Contract no.214898
Seventh Framework Programme: FP7-ICT-2007-1

List of the authors

Project acronym	MANCOOSI
Project full title	Managing the Complexity of the Open Source Infrastructure
Project number	214898
Authors list	Roberto Di Cosmo ⟨roberto@dicosmo.org⟩ Anne-Sophie Réfloc'h ⟨anne-sophie.refloch@univ-paris-diderot.fr⟩ Arnaud Laprévote ⟨arnaud.laprevote@edge-it.com⟩ Peter Van Roy ⟨peter.vanroy@uclouvain.be⟩
Workpackage number	WP7
Deliverable number	3
Document type	Deliverable
Version	2 – Revised on April 7, 2010
Due date	31/01/2010
Actual submission date	10/02/2010
Distribution	Public
Project coordinator	Roberto Di Cosmo ⟨roberto@dicosmo.org⟩

Contents

1 Introduction **1**

2 Mancoosi collaboration plan with FP6/FP7 projects **1**

2.1 QualiPSo 1

2.2 SELFMAN 2

3 Other collaboration actions **2**

3.1 Participation in the SSAI collaboration meetings 2

3.2 Workshops 3

3.3 During the current reporting period 3

3.4 Forthcoming 3

4 Collaboration with non EU-funded projects **4**

4.1 Eclipse P2 / Sat4J 4

4.2 University of Murcia / Gra2Mol 5

1 Introduction

In the course of the Mancoosi project, a series of collaboration opportunities have emerged, like the ones with the EU-funded Selfman project and with some national projects such as the ones funded via the Free Software thematic group of the Systematic competitiveness cluster in the Paris region in France.

This report essentially presents an update of the collaboration plan previously submitted as Deliverable D7.2 in January 2009, covering the potential liaison and co-operation activities with the other ICT projects under the WP2007/2008 objective "Service and Software Architectures, Infrastructure and Engineering". Exploitation of synergies between Mancoosi and the other projects will consist in participation to workshops, contributions to working groups, joint dissemination activities and the production of joint dissemination material.

At this moment, concrete results from cooperation efforts are showing up essentially in Selfman (with which we share partner UCL), and Qualipso (that reuses some of Mancoosi's results via the shared partner Mandriva), so we decided to remove from this plan mention of S-CUBE, DEPLOY and NESSI: this does not mean that we definitely give up any plan to collaborate with them, but that we consider such collaboration less likely, in the short timeframe left before the project ends.

2 Mancoosi collaboration plan with FP6/FP7 projects

2.1 QualiPSo

Project name	QualiPSo
Web site	`http://www.qualipso.org`
Start date	01/10/2007
Duration	4 years
Project type	Integrated project
Coordination	Engineering Ingegneria Informatica SpA (IT)

As stated on the project's Web site, "QualiPSo intends to define and implement the technologies, processes and policies to facilitate the development and use of open-source software components".

Mandriva is both a member of the Qualipso project and of the Mancoosi project, so that the results coming from Mancoosi about package metrics, and integrated in Doc4, are made available to the Qualipso consortium.

2.2 SELFMAN

Project name	SELFMAN
Web site	`http://www.ist-selfman.org`
Start date	01/06/2006
Duration	40 months
Project type	STREP
Coordination	Université catholique de Louvain

The goal of SELFMAN is to understand how to build large-scale distributed systems as self managing, to reduce the complexity of managing them. SELFMAN has built a practical distributed storage with a transactional interface, over a peer-to-peer network, that provides a scalable key-value store. Two implementations were done, Scalaris and Beernet, with different properties but using a similar transaction management algorithm. One SELFMAN implementation, Scalaris, won first prize in the IEEE International Scalable Computing Challenge 2008.

The collaboration between SELFMAN and MANCOOSI has now become more concrete. Using the results obtained on transactional replicated storage, we are planning to contribute to MANCOOSI in three different ways, which are currently under evaluation. One way is to provide package distribution where a package's files will be obtained from different peers instead of from a centralized server. In addition, different package versions differ only in some files, as different version of an article on the Wikipedia differ only on some paragraph. Using our experience with the Wikipedia, we could also store efficiently the different versions of software packages.

A second way to contribute to MANCOOSI is by creating a network of user-groups having similar system configurations on their machines. The goal is to identify when the new version of a package has been successfully installed by the majority of users on such group. This would result on a notification that would help the "paranoid" user that only wants to install software that has been tested by other users.

A third contribution, which is more challenging, is to build a decentralized repository of successful *installation paths* to help other users to solve their installability problems. A solver trying to decide which packages must be installed on a machine so as to guarantee software dependencies and avoid conflicts, needs to follow several installation paths until it reaches its goal, or until it fails and retries with a different paths. If some paths can be known successful from a decentralized repository, it would help the solver to efficiently find the solution for its concrete problem.

3 Other collaboration actions

3.1 Participation in the SSAI collaboration meetings

In addition to direct collaborations with the working groups set up by the other FP7 projects listed above, Mancoosi took part to transversal initiatives such as the SSAI collaboration meeting on FP6 and FP7 projects on objective 1.2 on June 10-11, 2009, by the European Commission in Brussels, in order to identify new collaboration and dissemination strategies.

3.2 Workshops

3.3 During the current reporting period

As foreseen in the description of work, we organised a first international workshop on Mancoosi-related issues in Amsterdam, in August 2009. This workshop, named IWOCE for *International Workshop on Open Component Ecosystems* was affiliated to ESEC/FSE, one of the major software engineering conferences, all the papers presented at the workshop have been refereed, and the five accepted have been published in the conference proceedings.

The workshop was intimately related to the topics of interest of Mancoosi (the ecosystem of packages in FOSS distributions) and has generalized them for a wider audience. Roberto Di Cosmo and Paola Inverardi have chaired it.

The web site can be found at `http://www.iwoce.org/`, and the program was as follows:

- Marianne Simonot and Virginia Aponte. A declarative formal approach to dynamic reconfiguration
- Davide Di Ruscio, Patrizio Pelliccione, Alfonso Pierantonio and Stefano Zacchiroli. Towards maintainer script modernization in FOSS distributions
- Daniel Le Berre and Pascal Rapicault. Dependency management for the Eclipse ecosystem
- Ralf Treinen and Stefano Zacchiroli. Expressing Advanced User Preferences in Component Installation
- Vasilis Boucharas, Slinger Jansen and Sjaak Brinkkemper. Formalizing Software Ecosystem Modeling

3.4 Forthcoming

A second Mancoosi workshop, LoCoCo, will take place in the summer of 2010, as a workshop affiliated to the prestigious FLOC reseach conferences. The FP7 projects mentioned above will be invited to submit contributions, as well as many other potential contributors from industry and academia.

4 Collaboration with non EU-funded projects

4.1 Eclipse P2 / Sat4J

As stated on the Eclipse website, *Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle.* Eclipse is a solid infrastructure on which sophisticated development environments can be built by extending the base platform by means of *plugins*, that play in Eclipse a similar role to packages in a GNU/Linux distribution. Managing hundreds or thousands of Eclipse plugins leads to problems similar to the one posed by managing hundreds or thousands of packages in a GNU/Linux distribution, thus providing an excellent basis for collaboration. As of Eclipse project build I20080305 (shortly before Eclipse 3.4/Ganymede M6), the Eclipse SDK contains a new provisioning system called Equinox/p2, maintained by Pascal Rapicault, that uses SAT solver technology from Sat4J to satisfy the dependencies when new plugins are installed.

Sat4J `http://www.sat4j.org/` is an open source project whose target is to make available modern SAT technology in Java, as a library reusable in other projects. Developed at the Centre de Recherche en Informatique de Lens (`http://www.cril.univ-artois.fr/`), by a team coordinated by Daniel Le Berre, it is used nowadays by a range of research and industrial projects. It can be downloaded through the OW2 forge.

From the beginning of the project, we kept in touch with Daniel Le Berre and Pascal Rapicault, and the most interesting convergence point turned out to be the optimization criteria: while the use of solvers for provisioning is no longer a new idea (it was suggested in the EDOS project back in 2005, and then implemented in various industrial systems, ranging from the Opium package manager for Linspire, to the more recent Suse package manager), little work has been done on the optimization criteria to be used when selecting one solution out of the possibly exponential solution space.

Significant evidence of the collaboration is the participation of the P2/Sat4J team in the internal competition that took place in January 2010: this internal run was meant to test and finalise all the infrastructure necessary to run the International competition, and was in principle reserved to the members of the project, but it turned out that the P2/Sat4J team had built solid knowledge of the CUDF format, at the heart of the competition, so they asked to participate, and we gladly welcomed them onboard. During this first internal run we uncovered a large amount of glitches in the infrastructure, so that the scores of the participants were incorrect for a long time, but it is remarkable that the solvers submitted by the P2/Sat4J team were the most correct and efficient at the moment of the run, even if, after fixing the errors in the infrastructure, and helping the other entrants to correct their errors in parsing the CUDF document, several other entrants got similar or higher scores, and one of them, INESC, reused part of the code from P2CUDF to achieve these results.

We plan to continue collaboration in particular on the design of optimization criteria well adapted to component based software systems: the different settings of GNU/Linux distributions on one side, and Eclipse plugins, or even Maven artifacts on the other side, carry the promise of obtaining results on general interest.

4.2 University of Murcia / Gra2Mol

In the context of the Work Package 2 a fruitful collaboration has been established with the research group of Prof. Jesús García Molina from the University of Murcia (Spain). In particular, from the beginning of the second year of the Mancoosi project, we kept in touch with Javier Luis Cánovas Izquierdo and Jesús Sánchez Cuadrado who developed and maintain Gra2MoL[1] which is a domain specific language tailored to address the problem of model extraction from source code. Gra2MoL has been adopted to implement the maintainer script injection procedure able to take existing package maintainer scripts and represent them as models specified as statements of the DSL proposed in the Deliverable D3.2. In particolar, given a metamodel, a AntLR grammar and a set of mapping rules which connects the entities in the grammars to those of the metaclass, a bridging mechanism is automatically obtained such that for each program generated by that grammar a model conforming to the metamodel is produced.

During the collaboration a number of bugs affecting Gra2MoL have been fixed and many new features have been introduced to increase the expressiveness of the language and to meet some requirements we had in WP2. The improvements which have been implemented in the new version of the tool enable the specification of complex source code queries and transformations which were not previously supported. We plan to continue the collaboration with Gra2MoL people which are very interested in applying their tool in Mancoosi which represents for them an appropriate and good play ground.

[1] Gra2MoL - Grammar to Model Language: `http://modelum.es/trac/gra2mol/`