

Project Presentation Deliverable D1.1

Nature : Deliverable

Due date : 01/08/2008

Delivery Date: 01/08/2008

Start date of project : 01/02/2008

Duration : 36 months

Distribution: Public





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
Author list	Roberto Di Cosmo Sophie Cousin
Workpackage number	WP1
Deliverable number	1
Document type	Deliverable
Version	1
Due date	01/08/2008
Actual submission date	01/08/2008
Distribution	Public
Project coordinator	Roberto Di Cosmo (roberto@dicosmo.org)

Contents

1	Overview of the project	5
1.1	Summary	5
1.2	Consortium Members	7
2	Concept and objectives, progress beyond state-of-the-art	9
2.1	Concept and objectives	9
2.1.1	Context	10
2.1.2	F/OSS: a complex, decentralised system	11
2.1.3	A fundamental challenge: managing the evolution of the F/OSS complex system	11
2.1.4	Objectives	13
2.2	Progress beyond the state-of-the-art	15
2.2.1	Indicators	18
2.3	S/T methodology and associated work plan	19
2.3.1	Overall strategy and general description	19
2.3.2	Work package 1: Project management	21
2.3.3	Work package 2: Models for the description of software artifacts and the upgradeability process	22
2.3.4	Work package 3: Transactional upgrades	25
2.3.5	Work package 4: Study and development of specialized upgradability solvers	27
2.3.6	Work package 5: International competition of constraint solvers for upgradeability problems	29
2.3.7	Work package 6: Dissemination	31
2.3.8	Work package 7: Collaboration with ICT SSAI&E Projects	33
2.4	Consortium as a whole	34
2.4.1	University Paris Diderot (JRU - Joint Research Units)	35
2.4.2	Edge-IT	36
2.4.3	University of L'Aquila	37

2.4.4	Caixa Mágica Software	38
2.4.5	INESC-ID	39
2.4.6	University of Nice-Sophia Antipolis (JRU - Joint Research Units)	40
2.4.7	Tel Aviv University	41
2.4.8	ILOG SA	42
2.4.9	University of Louvain	43
2.4.10	Pixart Argentina	44
3	Impact	45
3.1	Strategic impact	45
3.2	Plan for the use and dissemination of foreground	47

Chapter 1

Overview of the project

1.1 Summary

Project summary

Mancoosi aims to develop the scientific knowledge and build the tools necessary to manage the complexity of the open source infrastructure, which is one of the essential building blocks of tomorrow's software architectures: the success of LAMP (Linux, Apache, Mysql, Php) inside and outside the data centers is clear evidence of this. Yet, this infrastructure undergoes a fast-paced and distributed evolution that is too often maintained in ad-hoc ways using tools and processes that have clearly attained their limits today: we need new and innovative technology.

We explicitly target the difficult problems that arise when one wants to efficiently and safely upgrade a set of software components in complex software infrastructures, like those found in open source software distributions. These are among the most complex software systems known, made of tens of thousands of components that evolve over time without centralized design, and provide today a real-world example of what tomorrow's complex, quickly changing software systems will look like: technologies developed in Mancoosi will pave the way to the maintainability of the systems of the future.

Mancoosi will provide: a model of the infrastructure, and the transformations it undergoes when adding or removing components; advanced algorithms to choose efficient evolution paths when updating a platform; a forum to attract leading experts by organizing an international competition; tools that incorporate these findings and advance the state of the art in the field.

Mancoosi is precisely focused on Objective ICT-2007.1.2 (c) of the present call, by enabling mastery of complexity, dependability and behavioral stability in the complex system of software infrastructures evolving over time in a decentralized manner.

We bring together innovative industries from the Open Source world with deep first-hand experience in the practical issues of the problem, and a privileged channel for dissemination of the results; members of the Open Source community who are able to provide high-quality access to the community; and leading researchers who have the knowledge necessary to elaborate the

sophisticated models and algorithms needed to solve the underlying problems.

1.2 Consortium Members

Role	Number	Name	Short Name	Country	Date Enter	Date Exit
CO	1	University Paris Diderot	UPD	France	1	36
CR	2	Edge-IT	EDG	France	1	36
CR	3	University of L'Aquila	UDA	Italy	1	36
CR	4	INESC-ID	INI	Portugal	1	36
CR	5	Caixa Magica	CXA	Portugal	1	36
CR	6	University of Nice	UNSA	France	1	36
CR	7	University of Tel Aviv	TAU	Israel	1	36
CR	8	ILOG	ILG	France	1	36
CR	9	University of Louvain	UCL	Belgium	1	36
CR	10	Pixart	PXT	Argentina	1	36

Role: CO=Coordinator; CR=Contractor.

Chapter 2

Concept and objectives, progress beyond state-of-the-art

2.1 Concept and objectives

The main objective of the Mancoosi project is to develop the scientific knowledge and build the tools necessary to manage the complexity of the open source infrastructure. This infrastructure is one of the essential building blocks of tomorrow's software architectures: the success of LAMP (Linux, Apache, MySQL, Php) inside as well as outside the data centers is clear evidence of this. Yet, this infrastructure undergoes a fast-paced and distributed evolution that is too often maintained in ad-hoc ways using tools and processes that have clearly attained their limits today. There is a need for new and innovative technology, and this is what Mancoosi will provide.

We explicitly target the difficult problems that arise when one wants to efficiently and safely upgrade a set of software components in complex software infrastructures, like those found in open source software distributions, which are among the most complex software systems known, made of tens of thousands of components that evolve over time without centralized design: this is a challenging endeavor.

This goal makes the Mancoosi project precisely focused on Objective ICT-2007.1.2 (c) of the present call:

Service and Software Architectures, Infrastructures and Engineering.

Strategies and technologies enabling mastery of complexity, dependability and behavioral stability in complex systems and in systems evolving over time without central design.

We adopt a very focused strategy bringing together innovative industries from the Open Source world that have deep first-hand experience in the practical issues of the problem, members of the Open Source community that are able to provide high-quality access to the community, and leading researchers that have the knowledge necessary to elaborate the sophisticated models and algorithms needed to solve the underlying problems.

We have chosen the Open Source infrastructure as our main target, as it provides today a

real-world example of what tomorrow's complex, quickly changing software systems will look like: the applicability of these models and algorithms goes far beyond Open Source software, and technologies developed in Mancoosi will pave the way to the maintainability of the software systems of the future, especially for systems of systems, even when they are not Open Source.

In the following we will present in detail these issues and the organization of the project.

2.1.1 Context

Today's software systems are very complex modular entities, made up of many interacting components that must be deployed and coexist in the same context.

Modern operating systems provide the basic infrastructure for deploying and handling all the components that are used as the basic blocks for building more complex systems. These infrastructures are the very foundation of any distributed system, from the Internet to the ones Service Oriented Architectures are built on.

The world is moving towards an on-line mode of operation where most of the activities are performed through services that are accessed through the Net. Since users rely on these services to carry out their daily work, service providers must guarantee a high quality of service. Long downtimes due to maintenance problems could compromise the work of thousands of people and might cause big loss for the parties who manage the service itself, both in terms of money and reputation.

With the shift towards an on-line mode of operation the "minimal acceptable standard" for the quality of service of distributed systems has been raised to a very high level. We almost never see an *Error* page on Google Mail or on their search Engine. But when it happens, even temporarily for a single search, we already start to feel disappointed. Being unable to access an online service for several minutes might be considered unacceptable. For example, a Google search page with a banner saying "*Down for maintenance from 10am to 12am*" would be unconceivable. As small and medium enterprises rely more and more on these distributed systems for their daily businesses, this concern is not restricted to big corporations.

Any of these players in the modern information infrastructure is faced with the dilemma of

- on the one hand having to maintain their data centers and production machines in a healthy state, in order to guarantee the availability today's users expect,
- and on the other hand maintaining their systems in an always up-to-date state, not only because of the evolution in the core technologies, but especially to address security vulnerabilities that are found very frequently and must be corrected immediately in order to protect the system from malicious attackers that could compromise it with catastrophic consequences.

One of the most challenging environments where these issues arise is that of Free and Open Source Software (F/OSS) distributions. The F/OSS movement has gained — and is still gaining — momentum; applications developed by F/OSS communities are being deployed in many contexts, be it private users, education and research, public administration or professional Internet service providers. It is no surprise that big companies such as Google and Linden Lab, or large public bodies, like the french Ministry of Finance, build an important part of their information technology infrastructures relying on F/OSS components.

2.1.2 F/OSS: a complex, decentralised system

In the F/OSS world components evolve independently from each other. Development projects may be short-lived or longed-lived, have a central organizational structure or no organizational structure at all, may be conducted by a single person, a small homogeneous group of developers, an open group where contributors join or leave at will, or a large and geographically distributed group of developers. Projects may fall into a state of suspension, and suspended projects may be picked up and put back to live under a completely different organizational structure. As a consequence, project management, objectives, quality assurance procedures, version control, release process and contributors' goals and motivation vary widely.

While it is in principle possible to deploy a complete F/OSS infrastructure by fetching the sources and recompiling each components from scratch and independently, the complexity of this task is daunting, and has led to the development of what are now called *distributions*, which have a privileged place in the F/OSS market. This problem has become so complex that there are now even intermediate agents creating software bundles from different sources, which then in turn are taken up by distribution editors. Examples of these are the *CPAN* network of Perl libraries, or bundles of typesetting utilities for the \TeX system like *tex-live*.

A distribution is a consistent and functional collection of software components comprising all the software that is necessary to set up a system, in other words a complete operating system. Distributions may be general purpose or cater for a specific application domain. There are desktop-oriented distributions for the large public (e.g., Mandriva, Ubuntu, Fedora, etc.), server-oriented distributions for managing and running distributed systems providing services to the users, and even more specific ones targeted to mobile phone, home appliances and so on. Some general-purpose distributions like for instance Debian have spawned off domain-specific, so-called custom distribution, for instance for educational purposes (Skolelinux) or medical applications.

A distribution contains at least one kernel of the operating system (today often a GNU/Linux kernel and/or a BSD kernel, in the future we may see in addition the choice of a GNU hurd kernel), all the essential software components that are necessary to make a basic system operative (this is usually an implementation of the well-know UNIX tools), and a choice of user applications.

2.1.3 A fundamental challenge: managing the evolution of the F/OSS complex system

Typically a distribution has an automated mechanism for managing the components it is made of. These components, in fact, are often provided in a packaged form, i.e., in a format that can be easily processed by automatic tools, like `dpkg` and `rpm`, and that contains some additional information useful to handle their installation, removal and update. The most important information concerns the specification of inter-package relationships such as *dependencies* (i.e., what a package needs in order to be correctly installed and to function correctly), and *conflicts* (i.e., what should not be present on the system in order to avoid malfunctioning).

Because of the huge amount of F/OSS components (and packages) available, it is not easy to manage the life-cycle of a distribution; users are in fact allowed to choose and install a wide variety of alternatives whose consistency cannot be checked a priori to their full extent. It is then possible to easily make the system unusable by installing or removing some packages that “break” the consistency of what is installed in the system itself. In the case of commercial

operating systems where the core components are developed and controlled by a single entity this problem is either mitigated or partly hidden: it is not possible, for example, to change the boot loader or the graphical subsystem of Windows, as only Microsoft is allowed to do that consciously and hopefully safely; still, the dreaded *DLL hell* problem is fully present even if the user is too often left with the impression that any problem comes necessarily from a third party vendor.

Another problem consists in ensuring the correct upgrade of such a kind of system. This is an even more complex problem than the installation of single packages since we might have to preserve some additional properties and find an “optimal-path” to migrate the system from the current state to the targeted new state. Of course the basic property that we want to preserve is the consistency of the system. Since an upgrade is no more than a removal of some components and the installation of more recent ones, all the issues that apply to the installation of single packages are also valid in this context. There may, however, be additional constraints. For example, it would be reasonable to ask for an upgrade of the whole system that “preserves” the web-server infrastructure in its current state, (e.g., we don’t want our Apache 1.3 to be upgraded to the latest available version). Or, alternatively, it would be reasonable as well to ask for upgrades that minimize the size of the installed packages (e.g., in the case of limited devices or home appliances) or that remove the least number of packages (e.g., we would like to preserve as much as possible of the previous system).

This highlights the necessity of taking also into account non-functional aspects in the package selection and installation. With this respect, non-functional parameters in this setting can be partitioned into two categories: (i) environmental parameters - that are the ones affecting the downloading of packages, such as the availability and throughput of network connections to different sites hosting copies of the same package, (ii) system parameters - that are proper to the running system and are affected by the characteristics of different package versions, where a choice between different types of (functionally equivalent) packages is in place.

All these constraints complicate the management of the system, and it would be useful to have some tools that can handle this complexity by exploring the space of possible solutions and report to the users, in a succinct but meaningful way, the different possibilities and trade-offs. And, of course, these tools must also be able to enact the changes and to take the system to the new state correctly. Additional constraints shall be formulated to guarantee a certain level of Quality of Service upon installing new packages by taking into account non-functional aspects such as reliability, performance, etc.

Until now, these issues have been dealt with in ad-hoc ways. Maintainers of a server machine, anxious to satisfy very high availability expectations, play conservatively and use a “do not touch anything” approach, by keeping things that work in that state until modifications become unavoidable (e.g., to fix a severe security issue). Moreover, in critical systems, people often maintain and modify systems by hand, seldom relying on automatic tools. This is reasonable when the size of the server installations are small but becomes unacceptable when this size grows.

Software distributors strive to ensure the existence of safe upgrade paths between successive versions of their distribution, unfortunately with varying degrees of success. Distributions aiming at desktop users are under the pressure that they have to provide the latest “cutting-edge” software, and they are sometimes tempted to compromise quality assurance for the perceived benefit of being closest to the edge. Much too often this results in users following a “backup and redo from scratch” approach: upgrade by actually doing a fresh installation of the new system, often after having unsuccessfully tried to perform an “automatic update”. This may

be reasonable for a single desktop system not requiring a heavily customized configuration and for which a downtime of several hours is without consequence, but it is absolutely unacceptable in an enterprise context where there are hundreds of workstations.

Given the challenging nature of the F/OSS world and its widespread utilization, we want to focus on this context in order to provide advanced technology for handling the previously described scenarios. So our ultimate objective is to *improve the state of the art for upgrading and maintaining complex Free and Open Source platforms*, with a particular focus on GNU/Linux distributions.

2.1.4 Objectives

The main objective of this project is to give a significant contribution to the management of complex software systems that are built of components that evolve in a non centralized fashion, and that can be composed in unforeseen ways. Such scenarii are occurring in widely different frameworks that use component-based architectures.

One particular example of such component based architecture, which is extremely significant for its high complexity, large user base, and diversity of components is the Open Source infrastructure, and we want to improve its manageability by developing formal models, algorithms and tools to ensure that a complex Open Source infrastructure can always be kept in an up-to-date and consistent state, and keep up with the fast pace with which new components and technologies are released in the Open Source world.

Our strategy to reach this objective is twofold: on one side, we will build a formal model of the effects of the installation/removal of a software package on a system, that will enable us to conceive and implement a transactional update process, which is an essential component for ensuring maintainability of a system; on the other side, we will tackle the upgradeability problem.

The upgradeability problem we are attempting to solve can be stated as the following: we have a system in a state U , a set of resource S , and an update request R ; we want to bring U to a state U' that contains R using resources of S ensuring the U' is “correct” (i.e., nothing is broken with respect to the functionality provided by the system in the new state U'), and the request R is fully satisfied.

Upgradeability is an emerging problem present in various domains: it is a quite fundamental issue in GNU/Linux distributions, which have a variety of formats for representing the metadata of the components; it is present in various object component models, that use some other metadata to describe dependencies and conflicts; it also appears in the description of composition of services.

An upgradeability problem comes with its own cost and utility model, and is in general extremely complex to solve (in many cases beyond NP-completeness¹): it is of paramount importance to find *efficient* optimization algorithms that provide good approximate solutions to upgradeability and maintenance problems.

In this project, we specifically target the upgradeability problems that arise in F/OSS (Linux, *BSD, etc.) distributions for the following reasons:

¹NP-completeness is commonly considered as one of the most significant barriers separating tractable from intractable problems, yet approximate solutions can sometimes be efficiently found even for problems whose complexity is beyond NP.

- they embody the essence of a “complex” system that evolves with no central control, and they are rich of use cases that highlight the issues we are addressing;
- they are widely used either to power “server-side” infrastructures and “client-side” workstations. So, though the purpose of the usage is different, problems are almost the same; providing tools for handling these problems would have a big impact and they will be beneficial for a large spectrum of activities;
- they are used by millions of users worldwide, in a very demanding way, everyday: as such, they provide the potential to collect very significant experimental data to test and validate our proposed solutions (Milestones M5.2 and M5.3);
- F/OSS distribution provide today a real-world example of what tomorrow’s complex, quickly changing software systems will look like: technologies developed in Mancoosi will pave the way to the maintainability of the systems of the future.

The outcome of the project will be a framework that will comprise generic models for describing the state of a system (in particular of GNU/Linux systems) and a set of algorithms and tools that, operating on these models, will be able to provide to end users the means for handling, in way as automatic as possible, the state of the systems and to perform changes and upgrades on it (Milestones M2.2, M2.3, M3.2 and M4.4).

This corresponds precisely to the organization of the different workpackages:

- models of the system state which is transformed during upgrade operations (Workpackage 2);
- tools and algorithms to enact these operations in a transactional way (Workpackage 3);
- specialized optimization algorithms to discover efficient upgrade paths (Workpackage 4);
- an international competition specially designed to attract researchers to collaborate in improving these algorithms (Workpackage 5).

The models, tools and algorithms created by the Mancoosi project will actually lower the perceived complexity of the systems and provide to end users an effective and efficient way to manage their complexity in a simple and correct way (Milestones M2.3, M4.3).

The project is a highly innovative project targeting concrete issues, with a very high potential impact:

Innovation. The optimization and solution-search problems that must be solved have a high degree of scientific interest. Efficient solutions to optimization problems are a research topic which attracts a lot of research activity. One of the outcomes of the project will be the automatic generation of a large body of real-world and non-trivial optimizations problems. This will provide a base of use cases that can be used by researchers in the field to test their results. This will be to the benefit both for the progress of the research per se, and for the outcomes that this research will have in terms of techniques and technological advances in this context (Milestone M5.4).

Big Impact. Open Source infrastructures, and in particular, GNU/Linux systems, are widely used, with users ranging from big enterprise data centers to small and stand-alone desktop

systems. Having the technology and tools for managing the very foundational problems (i.e., maintaining and upgrading the system state in a correct way) will have a very real impact for an enormous “audience” (Milestones M2.3, M3.3, M4.3).

Focus on a concrete and frequently occurring problem. Despite all the active work done by the Open Source community, upgrade problems are still done in ad-hoc ways. There is little scientific work in this area and there currently are no satisfactory solutions to the upgradability problem that stand on a sound foundation. Nevertheless, there is a well-recognized need in the Open Source community for good solutions. By focusing on this problem, we are going to tackle issues that are considered fundamental, that are perceived as an important and difficult problem in the Open Source world, and we are going to provide cutting-edge solutions and define the new state of the art in this field (Milestones M4.2 and M2.2).

2.2 Progress beyond the state-of-the-art

State of the art in the management of package-based software systems and F/OSS distributions Some of the Mancoosi partners have collaborated in the framework of the EDOS FP6 project² which was devoted to the specific problem of *building* Linux distributions, and have built deep and significant knowledge of the issues related to package-based software systems. This has led, among the various other results of the EDOS project, to the development of ground-breaking tools that allow to efficiently check the consistency of a set of packages: these tools are described in [MBDC⁺06], and are today successfully adopted and used daily in mainstream Linux distributions ranging from Caixa Magica, to Debian and Mandriva.

It was clearly stated in the goals of the EDOS project that the focus there was on improving the stability of a distribution *from the point of view of the distribution editor*, and not the stability of a particular user installation, which is Mancoosi’s main target.

Indeed, the problem of maintaining a user installation is quite more difficult: one needs to take into account the evolution over time of the installation, the availability of software coming from different distributions, or even from third parties, as well as sophisticated user preferences.

A survey of existing tools and technologies for maintaining user installations shows clearly that no satisfactory solution is available today: in the field of component models, one often finds methodologies where the software programmer is explicitly requested to ensure compatibility of a component with all previous versions, which is an unrealistic expectation when building very complex systems. In the field of Linux distributions, tools like `apt-get`, `urpmi`, `smart`, `synaptic` and others lack a lot of important features: efficient dependency solving, roll-backing, distributed retrieving of packages and better mechanisms for hardware drivers distribution. These tools do not have rollback mechanisms - with the exception of Apt-get through the preliminary Caixa Magica contribution - and have poor fault-recovery support. Distributed retrieval of packages, for instance using a peer-to-peer network, is not realized in today’s meta-installers.

²<http://www.edos-project.org>

Regarding dependency solving, which is probably the most urgent problem in the short run, different approaches exist, but all the mainstream tools are either incomplete (unable to find an upgrade path when it exists), explosive (they do not give an answer in acceptable time), or both.

No serious attempt at solving these problems using advanced problem solving techniques from academia has been made before, even if some interest on these issues is starting to build up as a consequence of the success of EDOS in the related, but different, problem of checking consistency of components.

Towards a new generation of tools By pulling together academia, industry and user communities, Mancoosi will provide advanced algorithms and tools that will constitute significant progress with respect to the existing technologies. We will incorporate state-of-the-art problem solving technology into a new generation of meta-installers (the common name given to tools like the ones surveyed above). On a first phase, we envision applying existing state-of-the-art solvers (e.g. MiniSAT, MiniSAT+, Pueblo, bsolo) to the upgradability problem. On a second phase, we plan to further develop existing solvers or even developing new solvers. Such new solvers should extend the ideas in SAT/PB solvers to naturally deal with the the upgradability problem.

We will propose a common upgradability description format (CUDF) for describing instances of upgradability problems, and of their solutions. CUDF will comprise the basic elements of common package description formats (like Red Hat's RPM format or debians deb format) that have already been identified in the EDOS project like unit name, version number, dependencies and conflicts. The format will only contain those elements that are essential for computing upgrade paths and will abstract from details that are specific to a particular distribution infrastructure. The purpose of this format is to evaluate the capacity of solvers to find upgrade paths, the format is not intended to replace existing package description formats. In particular, this format will be used in the solver competition which will be organised by the project. Solvers that are extended, or newly developed, by us in this project should be able to read directly CUDF problem descriptions, and therefore have dedicated data structures, as well as to receive information about which packages are more critical to update and to incorporate this information into the decision heuristic.

We will also propose a new generation of tools for managing the configuration of complex systems, that will allow a more refined description of the state of a system as it is altered by package installation or removal.

Models will play a key role here since they permit to have rigorous descriptions of the current state of systems and hence to enable different kinds of analysis and verifications. For instance, with respect to the abstraction of the system provided by the models, it will be possible to perform "upgrades from external sources", that is to inspect the state of the system and to understand what it is needed by an unpackaged software in order to be deployed on the system. The specification of precise models that can be automatically analyzed and manipulated requires the definition of metamodels, i.e. the precise definition of the abstract syntax of models and the interrelationships between model elements.

Up to now, *metamodeling formalisms and technologies* (for instance OMG's Meta-Object Facility [OMG03], Eclipse Modeling Framework [Ecl05]) have never been used to support solutions for the upgradability problem. In this project, such technologies will be exploited in order to

formally define metamodels that capture the most used and relevant information that is used while managing complex systems. Moreover, the project will provide a representation of changes from a model version to another in order to isolate possible “failure points” and conflicts. We believe that the outcomes of this study will benefit the overall area of *model versioning*. In fact, today’s available approaches supporting model difference/evolution are often limited to UML and proprietary representations. In a first phase of the project, such techniques will be improved in order to represent model differences in general; then they will be used to reflect the evolution that a system undergoes throughout its life-cycle.

State of the art in problem solving technologies We plan to further develop existing solvers to naturally deal with the upgradability problem. As a side effect, this will lead us to advance the state of the art in the research field of problem solving.

Two different techniques for solving complex combinatorial problems are at the basis of this project, namely Constraint Programming and Boolean Satisfiability.

Constraint programming (CP) is a programming paradigm which integrates concepts and techniques from artificial intelligence, operational research and logic programming. Constraint programming is an embedding of constraints in a host language. The first host languages used were logic programming languages; the field was consequently in the beginning called constraint logic programming. The two paradigms share many important features, like logical variables and backtracking. Today most Prolog implementations include one or more libraries for constraint logic programming. The first implementations of constraint logic programming were Prolog III, CLP(R), and CHIP. The most popular constraint logic programming interpreters are GNU Prolog, ECLiPSe (open source), and SICStus (proprietary). Constraints can also be embedded in an imperative language via constraint solving toolkits, which are separate libraries for an existing imperative language. ILOG CP and Gecode are examples of such constraint programming libraries for C++. Constraint programming has become a mature technology. ILOG CP is used in numerous applications as for instance scheduling, routing, or car sequencing. Constraint programming is still an active domain of research with dedicated international conferences and journals.

Boolean Satisfiability (SAT) is the problem of finding a truth assignment that satisfies a Boolean formula, or proving that such assignment does not exist. SAT was the first problem to be proved NP-complete in 1971, and for that reason it is since a long time a very well known problem from a theoretical point of view. The first significant algorithmic approaches for solving the SAT problem were proposed by Davis and Putnam in 1960 (DP) and further developed by Davis, Logemann and Loveland in 1962 (DPLL). The former algorithm is based on resolution and requires exponential space in the worst case, whereas the latter is based on backtrack search, thus avoiding a memory explosion. However, only in the 90’s the first effective algorithms for solving real world problems were introduced with GRASP and relsat. These algorithms incorporate learning techniques to avoid repeating the same mistakes and allow performing non-chronological backtracking. Later on, the Chaff SAT solver incorporated lazy data structures and a dynamic branching heuristic, which turned out to make SAT solvers even more efficient. Currently, SAT solvers are used in different domains including hardware and software verification, automated deduction, FPGA routing, planning, knowledge discovery and bioinformatics. Furthermore, SAT is nowadays interpreted in a broad sense: besides Boolean satisfiability, it includes the domains of pseudo-Boolean constraint solving and optimization (PB), quantified boolean formulae (QBF), constraint programming techniques (CSP) for word-level problems and their propositional encoding, and satisfiability modulo theories (SMT).

The integration of Constraint programming techniques and SAT techniques is a hot topic. SAT and CP techniques are two problem solving technologies which share many similarities. The techniques used in SAT (propagation, activity-based heuristics, conflict analysis, restarts...) constitute a very successful combination which makes modern DPLL solvers robust enough to solve large real world instances. Not all problems are nevertheless effectively expressed in a Boolean format, and one limitation of SAT solvers is that they provide no native support for constraints on numeric or other non-Boolean domains. There is therefore considerable interest in cross-fertilizing SAT techniques with techniques developed for more general problems in the Constraint Programming literature.

2.2.1 Indicators

Our project clearly focuses on Objective 1.2 (c): Strategies and technologies enabling mastery of complexity, dependability, and behavioural stability.

The overall indicators for the target outcome, and their target objectives at the end of the project are indicated in the following table:

Indicator	Target
Peer reviewed articles	6 in international conferences like ASE, CP, SAT, ICSE, etc.
Open source repository	1 reference repository for the development of the tools, with a user community of several thousands
International Workshops	2 associated to international conferences like CP, SAT, ASE, ICSE, etc.
Dissemination	2 articles in user journals like Linux Journal, etc.

The overall impact indicators, and their target objectives at the end of the project are indicated in the following table:

Indicator	Target
Open and standard platforms and interfaces	integration of an instance of the installation tools from Mancoosi in at least a mainstream distribution for each of the .deb and .rpm package formats.

2.3 S/T methodology and associated work plan

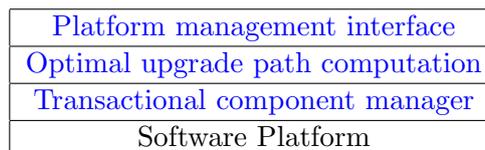
2.3.1 Overall strategy and general description

Let's recall that the project is clearly focused on advancing the processes and tools necessary to master the evolution of an open source platform: we want to make sure it is possible to keep it always up-to-date and in a consistent state, by adding and/or removing components in an optimal way.

Our strategy to reach this objective is twofold: on one side, we build a formal model of the effects of the installation/removal of a software package on a system, that will enable us to conceive and implement a transactional update process, which is an essential component for ensuring maintainability of a system; on the other side, we will tackle the upgradeability problem using advanced techniques from constraint optimization and boolean satisfiability.

The following picture gives an overview of the overall integrated architecture we will build as an outcome of the Mancoosi project: a user that wants to update or maintain their software platform will

- use a platform management interface that will allow them to specify in various ways their preferences about the new state (these may be functional preferences, like a mail service or a web service, as well as non functional preferences like size, speed, stability or quality of the updates);
- an optimization layer will compute upgrade paths, in the form of sets of components to replace, install or remove, according to the user preferences;
- the transactional component manager will perform the actual installation, removal or replacement of components, saving or restoring configuration information as appropriate.



We have accordingly split the work in Mancoosi among two main sets of workpackages:

- the definition of formal models of a software platform's state which is transformed during upgrade operations (WP2) and the development of a transactional update engine (WP3);
- the design of algorithms and tools for the discovery and optimization of upgrade paths (WP4), with the help of a large research community mobilized through an international competition that will multiply the impact of the project (WP5); competitions like the one we will organize have already shown to be extremely successful in attracting leading researchers and dramatically improving automated tools [MZ07, Sut07, SAT]. The competition should be organized in conjunction with a major international conference like for instance the International Conference on Theory and Applications of Satisfiability Testing (SAT), the International Conference on Principles and Practice of Constraint Programming (CP), or the International Conference on Automated Planning and Scheduling (ICAPS).

To these workpackages, we have added a workpackage for the management of the consortium and the liaison with the commission (WP1), an a workpackage devoted to dissemination (WP6).

2.3.2 Work package 1: Project management

Work package leader: University Paris Diderot

Objectives:

The objectives of this work package are the following:

- to initiate, manage and administer the project.
- to ensure the timely progress of the project with respect to the workplan and contractual commitments.
- to ensure the scientific and technical quality of the work.
- to provide an appropriate communication infrastructure for the project.
- to provide a communication channel between the consortium and the European Commission.

Description of work:

This workpackage is concerned with project management and therefore includes the work of the Project Management Board and Technical Committee as well as the project's co-ordinator and administrative co-ordinator.

The project co-ordinator will provide overall project management in consultation with the two controlling bodies, the Project Management Board responsible for all policy and strategic management matters and the Project Technical Committee responsible for technical and operational issues related to workpackages.

Deliverables:

Name	Due date	Description
D1.1	t0+6	Project presentation
D1.{2,3}	(t0+12, t0+24)	Progress periodic report
D1.4	t0+36	Final report, including report on awareness and wider societal implications

Milestones:

Name	Due date	Description
M1.1	t0+6	Project presentation
M1.2	t0+12	End of the first reporting period.
M1.3	t0+24	End of the second reporting period.
M1.4	t0+36	End of the last reporting period and final report.

2.3.3 Work package 2: Models for the description of software artifacts and the upgradeability process

Work package leader: University of L'Aquila

Objectives:

A model describing a complex system and the interdependencies occurring among its assets can be of crucial relevance for the management of system configurations. In fact, it permits to analyze rigorous descriptions to verify that configuration scripts provided with packaged components can safely be performed with respect to the abstraction of the system provided by the model. Also, the evolution and the modifications the system underwent during its life cycle can consistently be recorded. Models can be used to perform “upgrades from external sources”, that is to inspect the state of the system and to understand what is needed by an unpackaged software in order to be recompiled and installed in the system.

The specification of precise models that can be automatically analyzed and manipulated requires a precise definition of the abstract syntax of models and the interrelationships between model elements. In particular, metamodeling allows the construction of a collection of concepts within a certain domain formalized in a metamodel which describes the common properties of its instances according to the underlying semantics. A model is said to conform to its metamodel like a program conforms to the grammar of the programming language in which it is written.

A complex system may be composed of many loosely coupled machines. Similarly, knowledge about the properties of software pieces may be distributed among various external sources. A flexible model that accounts for such distributed scenarios allows for comprehensive system management.

By using well known metamodeling formalisms and technologies (e.g., OMG’s Meta-Object Facility [OMG03], Eclipse Modeling Framework [Ecl05]), in conjunction with distributed P2P data management technology, one of the objectives of this work package is to build a metamodel that enables the modeling of the relevant aspects of a complex system. In order to identify the elements and artifacts that should be modeled, the collaboration between Caixa Magica and UDA will elicit use cases that capture the most relevant information that is used while managing complex systems. Since it is not possible to specify in detail every single part of systems, trade-offs between model completeness and usefulness must also be evaluated in order to have a good model that fits well with them. Systems will be described under software and hardware perspectives even though, due to inherent limitations in hardware description and metadata, a limited set of drivers/hardware will be taken into account. The reference platform will be GNU/Linux systems where a high degree of complexity is already present, and which constitute a challenging target to be investigated.

The models that conform to the metamodel above should be “versionable”, i.e., models should be “diff-friendly” and it should be easy to represent changes from a version to another in order to isolate possible “failure points” and conflicts. Current approaches to model versioning are often limited to UML and proprietary representation, thus this work package will provide the support for representing model differences that will be taken into account by WP3 to manage the evolution that a system undergoes throughout its life-cycle. The evolution of the system must necessarily reflect how configuration scripts adopted

while installing or removing a package affect the overall state of the system. Thus, one of the goals of this work package is to investigate how to keep both the system and the model synchronized. In other words, the model describing the system after a package installation or removal is automatically obtained by a model transformation which captures the modifications due to the package configuration script.

Description of work:

Task 2.1 Find use cases that capture the most used and relevant information that is used while managing complex systems. In particular, given the standard life-cycle of a GNU/Linux system, identify what are the elements and artifacts that should be modeled.

Task 2.2 Develop a metamodel that captures the previously identified elements and that allows a reasonable description of the state of the system and the representation of its evolution over time, taking also into account non-functional aspects. Particular attention will be dedicated to building the metamodel in an adaptable and extensible way, so that it could be used in other contexts besides the reference one.

Task 2.3 Instantiate the metamodel on a GNU/Linux distribution (e.g., Mandriva) and develop a set of tools supporting the querying and manipulation of models.

Task 2.4 Develop a prototype for integrating the metamodel with system configuration/management tools (e.g., extensions to the package management system that takes into account the information provided by the model of the system before doing actual operations). This prototype will actually implement a framework that can be deployed and used for handling GNU/Linux distributions. Though the framework is specific for this kind of environments, an effort will be done to make it as generic as possible by clearly identifying and separating the parts that can be reused even in other environments that are specific to GNU/Linux distributions.

Deliverables:

Name	Due date	Description
D2.1	t0+12	Metamodel for describing system structure and state.
D2.2	t0+24	Instantiation of the metamodel on a wide-used GNU/Linux distribution.
D2.3	t0+36	Model-based framework for managing the complexity and the state of the GNU/Linux instantiation.

Milestones:

Name	Due date	Description
M2.1	t0+12	First version of the metamodel.
M2.2	t0+24	First version of the model for a given GNU/Linux distribution
M2.3	t0+36	Final version of the framework for a given GNU/Linux distribution and validation.

2.3.4 Work package 3: Transactional upgrades

Work package leader: Caixa Magica

Objectives:

The main objective of this package is to provide the technology for handling system evolution in a correct and safe way. By using the model developed in work package 2, the goal is to provide tools and algorithms to keep track of the evolution of the system and to revert the system to previous (working) states and retrieve it in an efficient way (for instance, using if applicable P2P technology).

We want to develop a transactional model of upgradeability that will allow to roll-back some parts of a possibly long upgrade history, restoring the relevant part of the previous configuration. Roll-back should then not only support the possibility of safely fall-back on the previous configuration of the system in case the upgrade process is interrupted, but the possibility of rolling back at any time in the future.

Three different levels of changes can be monitored: changes introduced by the copy of the files included in the new version of the package, changes introduced by the installation scripts and changes introduced by the user or by the applications after the installation.

As mentioned above, in the general case complex systems may be composed of many loosely coupled machines and we will strive to account for such a distributed setting.

The transactional update framework may be further extended to provide a better hardware support: drivers may be included in the packages and package constrains provide a mechanism for triggering a new driver installation when new hardware is added or removed. Since a complete hardware support would depend on having specific metadata on every hardware related package, WP3 will only address a testbed with a limited set of drivers.

Description of work:

Task 3.1 Identify the basic operations that can be used to make a GNU/Linux system evolve (i.e., atomic operations that can be composed to realize a complex upgrade). Study the state of the art technologies for transactional systems.

Task 3.2 Study compositional properties with respect to these operations, and develop a DSL (domain specific language) that can be used to describe control flows that can be used to represent complex system changes.

Task 3.3 Develop tools to keep track of the changes during time, in order to be able to roll them back, restoring previous safe system states.

Task 3.4 Integrate these tools in the framework described in work package 2 and with the utilities developed in work package 4.

Deliverables:

Name	Due date	Description
D3.1	t0+6	Survey of the state of the art technologies for handling versioning, rollback and state snapshot in complex systems.
D3.2	t0+12	First version of the DSL based on the model developed in WP2.
D3.3	t0+36	Final version of the rollback component that will be integrated in the framework developed in WP2.

Milestones:

M3.1	t0+12	First version of the DSL based on the model developed in WP2.
M3.2	t0+24	First version of the rollback component that will be integrated in the framework developed in WP2.
M3.3	t0+36	Final version of the rollback component that will be integrated in the framework developed in WP2.

2.3.5 Work package 4: Study and development of specialized upgradability solvers

Work package leader: University of Nice

Objectives:

This workpackage is in charge of developing tools and algorithms to solve the upgradability problem based on state-of-the-art knowledge coming from the academic partners.

Different state-of-the-art techniques could apply to tackle the upgradability problem with more or less success. We want to be able to test a large number of optimization algorithms and tools on instances of upgrade problems, to capitalize the experience coming from the research community. A modular platform is a key issue in this process : such a flexibility is a requirement to allow a quick integration and evaluation of a wide range of different possible techniques and algorithms on actual instances of upgrade problems, that will be provided partly by WP5. A significant part of the efforts of the partners will be devoted to the development of the Mancoosi modular platform.

The Mancoosi tools will also benefit from the research efforts of the partners to develop dedicated algorithms. Though different techniques might pretend to solve the upgradability problem, a subtle tradeoff between efficiency and the quest for a proper solution is required to give the right answer to this problem. Thus, the design of a "fine grain" cooperative framework for the different solvers is required to handle large systems of heterogeneous constraints. Exchange of failure information is a critical issue in such a framework. To achieve this aim, we will develop new dedicated algorithms for the upgradability problem.

Finally, the competition organized through WP5 will give us the opportunity to compare our tools and algorithms at an international level. This competition is a mean to evaluate our work in a challenging context and to gather new results and ideas to improve the Mancoosi tools and algorithms.

The objectives of this workpackage are highly challenging. The success of WP4 will rely on the important involvement of the Mancoosi partners.

Description of work:

The workpackage is organized around the following tasks:

Task 4.1 develop a tool, based on the Common Upgradeability Description Format (CUDF) defined in WP5, that is able to check a set of proposed solutions to an upgradability problem, and compute their actual cost, utility and quality values;

Task 4.2 develop a modular prototype platform manager able to apply various solvers to different upgradability problems described in the CUDF format and report the solutions found;

Task 4.3 design algorithms to solve upgradability problems based on the state-of-the-art in research and develop a prototype solver incorporating the algorithms found;

Task 4.4 take part to the competition organized by the WP5.

Deliverables:

Name	Due date	Description
D4.1	t0+18	First version of the Mancoosi modular platform manager.
D4.2	t0+24	First version of the Mancoosi specialised CUDF solver for the modular platform manager.
D4.3	t0+36	Final version of the algorithms and tools.

Milestones:

Name	Due date	Description
M4.1	t0+12	Checker for solutions to upgradeability problems.
M4.2	t0+24	Specialised algorithms and solvers.
M4.3	t0+36	Validation.

2.3.6 Work package 5: International competition of constraint solvers for upgradeability problems

Work package leader: University Paris Diderot

Objectives:

The main objective of this work package is to capitalize the experience coming from the research community, to attract researchers from all over the world to study this problem, and to foster development of more advanced tools. For this, we will set up an infrastructure allowing to test a large number of optimization algorithms and tools on instances of upgrade problems, and run an annual international competition. Running competitions of automated tools is a well established practice that has effectively promoted great improvements in tools in different areas over the past years, for instance for the termination of computational systems [Mar07, MZ07], theorem provers [Sut07, PSS02, SS06], and SAT solvers [SAT].

Description of work:

The workpackage is organized around the following tasks:

Task 5.1 Define CUDF (Common Upgradeability Description Format), a common, standard format for describing instances and solutions of upgradeability problems, in collaboration with major actors in the industry

Task 5.2 Develop extensions to commonly used upgradeability and management tools to collect difficult instances and solutions of the upgradeability problems, and represent them in the common format

Task 5.3 Build an Upgradeability Problems Data Base (UPDB) by lobbying the industry and communities of users to adopt the extended tools, and accept to

- Provide data from their actual failed upgrades (mention the similar successful experience of popcon in Debian);
- Produce additional data by running dry-run upgrades, that do not modify the user's system, of randomly selected popular components with respect to their actual installation

Task 5.4 Organize an international competition of constraint solvers that will allow to confront advanced research algorithms with the actual problems in the data base; this competition may be organized as a joint event with existing main international conferences.

Deliverables:

Name	Due date	Description
D5.1	t0+9	Description of the CUDF format.
D5.2	t0+30	Extension of an existing package manager to produce traces of upgradeability problems in CUDF format.
D5.4	t0+30	UPDB infrastructure to collect traces of upgradeability problems in CUDF format.
D5.4	t0+36	Report on the international competition.

Milestones:

Name	Due date	Description
M5.1	t0+9	CUDF format for describing upgradeability problems.
M5.2	t0+15	First version of the tools needed to create the UPDB
M5.3	t0+24	First international competition
M5.4	t0+36	Final validation

2.3.7 Work package 6: Dissemination

Work package leader: Edge-IT

Objectives:

The objectives of this workpackage are effective dissemination of the vision and of the achievements of the Mancoosi project through the following channels:

- A dedicated web site to promote the models and the tools.
- The organisation and promotion of a mid-term and of a final workshop.
- Research papers from the University partners.
- Conference talks.
- The integration of the Mancoosi tools in the Mandriva, Caixa Mágica and Pixart distributions.
- A dissemination report at the end of every reporting period.

Description of work:

Task 6.1 The consortium will participate in coordination events on F/OSS.

Task 6.2 The consortium will produce a dissemination report at the end of every reporting period. It will comprise web site access analysis, list of submitted research papers and appearance at conferences.

Task 6.3 Two workshops will be held at month 18 and month 36. Their subjects will be defined as the project moves forwards. The aim will be to promote interest in the community for the researched topics and technologies. We provide here a non limitative list of international conferences that would provide an adequate forum for the workshops; on the constraint solving side, some relevant possibilities are CP (International Conference on Principles and Practice of Constraint Programming, link to the last conference : <http://www.cp2007.org/>); CPAIOR (International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, link to the last conference <http://www.cs.brown.edu/sites/cpaior07/Welcome.html>); SAT (International Conference on Theory and Applications of Satisfiability Testing, link to the last conference : <http://sat07.ecs.soton.ac.uk/dispatch.php?disp=sat07>). On the software engineering side, some relevant possibilities are ICSE (International Conference on Software Engineering, <http://www.icse-conferences.org/>), FSE (ACM Symposium on Foundation of Software Engineering, <http://www.cc.gatech.edu/conferences/fse16/pastsoft.html>), ESEC (European Software Engineering Conference, <http://www.idt.mdh.se/ecec-fse-2007/index.php?choice=related>) and ASE (IEEE International Conference on Automated Software Engineering, <http://ase-conferences.org/ase/>

past/). As for the Open Source community at large, events like Linux-Tag, Fosdem or Linux World would be appropriate. Due to the organization structure of these events, it is not possible to determine so long in advance which one will be the actual forum for the workshop.

Task 6.4 The WP2 package management extensions, the WP3 rollback component and the WP4 CUDF solver will be integrated into the Mandriva Linux and the Caixa Mágica distributions.

Deliverables:

Name	Due date	Description
D6.1	t0	Creation of the project website.
D6.{2,3}	t0+12, t0+24	Reports on the outcome of the dissemination activities.
D6.4	t0+36	Final packages of the Mancoosi tools for Mandriva Linux and for Caixa Mágica.
D6.5	t0+36	Final plan for using and disseminating knowledge.

Milestones:

Name	Due date	Description
WP6-M1	t0+18	First workshop
WP6-M2	t0+36	Final workshop.

2.3.8 Work package 7: Collaboration with ICT SSAI&E Projects

Work package leader: EDGE-IT

Objectives:

This collaboration task covers the liaison and co-operation activities with the other ICT projects under the WP2007/2008 Objective “Service and Software Architectures, Infrastructure and Engineering.

The cooperation aims at exploiting synergies between the projects and increasing the impact of the ICT initiative.

Description of work: The consortium members commit to provide contributions to the following activities:

- Exploitation of synergies / technical concertation: : participation to workshops, contribution to some of the working groups.
- Joint activities for exchange, dissemination and training.
- Production of dissemination material that can be used for communication towards the general public.
- Contribution to repositories of reference implementations (details to be discussed)

This Task only covers the specific activities that will be done in collaboration with other projects. The other project workpackages cover the individual project activities in some of these areas (e.g., dissemination, standardisation).

Task 7.1 elaboration of the detailed collaboration plan

- at kickoff: the consortium discusses and agrees a way of working, which may start from the existing working groups
- at M6: the final detailed planning for collaboration is delivered

Deliverables:

Name	Due date	Description
D7.1	T0+6	Collaboration activities plan.
D7.{2,3,4}	T0+12, T0+24, T0+36	Report of activities done and update of the plan.

2.4 Consortium as a whole

To achieve Mancoosi's goals, we have to blend together a range of actors from industry, academia and user communities.

Let's recall that the project is clearly focused on advancing the processes and tools necessary to master the evolution of an open source platform via the definition of formal models of a software platform (WP2) and transactional updates to a software platform (WP3), via the design of algorithms and tools for the discovery and optimisation of upgrade paths (WP4), and via the mobilization of a large research community through an international competition that will multiply the impact of the project (WP5).

To make our work groundbreaking, we will collect large sets of data containing actual, real-world descriptions of update failures and maintainability problems, on an unparalleled scale, that will attract international researchers to this novel problem. This needs direct cooperation from distribution editors and their users; for this, we have given a prominent role in the consortium to two fully European Linux distribution editors: EDGE-IT, subsidiary of Mandriva, the leading European distribution editor, who has a very large installed base, and Caixa Magica, the leading distribution in Portugal. We will also secure direct cooperation with the Debian community through an active Debian developer who works for one member of the consortium, and through the Pixart distribution, which is Debian-based. The tools used by these companies and community cover practically all major variants of the existing ones, and provide exceptional first-hand experience. This same set of industrial partners will be paramount in ensuring massive dissemination of the project's results.

For our work to advance the state-of-the-art, we have secured participation of world leading experts in formal models (Università de l'Aquila), in distributed data query and management (Tel Aviv University) and in constraint solving, both from academia (Université de Nice, Université de Paris Diderot, Université Catholique de Louvain have complementary experience in constraint solving and formal methods), from industry (ILOG) and private research bodies (INESC-ID).

These partners have each unique experience in their respective fields, and the consortium is coordinated by University Paris Diderot who has shown in previous projects its ability to blend together academia, industry and the Open Source community.

Some of these partners have already cooperated very successfully in previous projects, and most notably the EDOS FP6 project.³ This constitutes excellent starting grounds for the success of Mancoosi.

³<http://www.edos-project.org>

2.4.1 University Paris Diderot (JRU - Joint Research Units)

**Organization description:**

University Paris Diderot is the leading multidisciplinary university in France, with teaching and research activities covering science, medicine, linguistics, literature and social sciences. Its focus has always been on excellence in research.

Scientific competencies and specific skills for this project:

The Department of Computer Science of University Paris Diderot is composed of two research laboratories which bring together world-renowned researchers in different aspects of formal methods in computer science:

PPS (UMR 7126) is specialized in topics including programming language design and foundations, formal methods, logic, semantics and proof of programs. PPS hosts a total of approximately 60 researchers, including about thirty permanent researchers, from five different countries.

LIAFA (UMR 7089) is specialized in topics including algorithms, combinatorics, automata theory, discrete events, specification and verification. LIAFA hosts a total of approximately 70 researchers, including about forty permanent researchers, from ten different countries.

Both laboratories have a long standing experience in Open Source Software, validation of complex software systems, and the creation of sophisticated tools to implement the theoretical algorithms that are the core of their research.

Contribution to the project:

Members of the two laboratories will actively contribute to the project by working essentially on workpackages 4 and 5 which require experience in designing optimized algorithms based on formal methods. One of the team members has a long standing experience in SAT-based solvers and is well integrated in the community, so there is a concrete possibility to efficiently publicize the solver competition in the community. Other members of the team have strong connection with the Open Source community, which will make it possible to foster adoption of the project's tools in a wide community. University Paris Diderot will also be the project leader; it has previous experience in collaborating and leading EU projects and has hired an assistant manager in order to efficiently coordinate the consortium.

2.4.2 Edge-IT



Organization description:

Edge-IT is a French SME fully owned by Mandriva, the editor of easy-to-use Linux systems for desktop or server architectures. Edge-IT is the support and consulting center of Mandriva products and focuses on integrating open-source software into complex industrial systems. In order to guarantee an adapted service regardless to circumstances, Edge-IT also provides a professional accompaniment (Consulting, Integration, Architecture, Audit). Edge-IT has a wide knowledge of several Linux distributions, including the Mandriva products.

As a reference support provider of Linux systems, Edge-IT is involved in several large multi-country, multi-competency European projects. Major projects include EDOS⁴ – F/OSS engineering and testing improvements, Ucopia – advanced mobility, NEPOMUK – social semantic desktop technologies and QualiPSo – quality platform for open-source software.

Scientific competencies and specific skills for this project:

Edge-IT has an extensive know-how on managing complex open-source projects. Edge-IT together with Mandriva maintains a series of tools such as URPMI for easing the management of package dependency on the client side. URPMI is a package management program that wraps around the package manager RPM with the aim that the user does not have suffer the often encountered “dependency hell” when installing software. Edge-IT will take advantage of its synergy with the Mandriva editor for involving the Mandriva Linux community into Mancoosi: the Mandriva community is among the most active communities of contributors to a Linux distribution, eager to integrate better dependency tools into the Mandriva Linux project tool chain.

Contribution to the project:

Edge-IT’s know-how in producing and maintaining software packages will be instrumental in the activities revolving around the modeling the software artefacts, the upgradability process, and for designing a transactional package manager. Through its involvement into the European Integrated Project QualiPSo, Edge-IT will build upon the general QualiPSo models and ontologies capturing the artefacts of F/OSS projects. Edge-IT will use the well established channels of Mandriva, with a large and international community of users and contributors for disseminating widely the achievements and the vision of the Mancoosi project together with all the partners. Edge-IT will lead WP6.

⁴<http://www.edos-project.org>

2.4.3 University of L'Aquila

Universita'



dell'Aquila

Organization description:

The Computer Science Department at University of L'Aquila (UDA) has developed a solid research and development experience in software engineering. The main focus of the research activities are software architectures, model-driven development, component-based programming, internet-based programming, security and verification issues. It collaborates with different companies in the telecommunication area and up to 2007 UDA has been involved and is involved in several national and international projects among which the MURST (www.miur.it) co-funded project SAHARA, FUTURE – IST-2000-25355, SAILOR – IST-2001-37266, PLASTIC – IST-2006-026955, POPEYE – IST-2006-034241, ARTDECO – Italian project - FIRB 2005.

Scientific competencies and specific skills for this project:

UDA has developed a consolidated experience in the field of metamodeling, model manipulation and analysis. Regarding model transformation and weaving, the undertaken research concentrates on the specification of transformation and of the weaving operators, and their application to several problem domains, such as web engineering, software architectures, and middleware-based communication systems. More recently, the research focus has been shifted on model difference, versioning, and evolution.

Contribution to the project:

Members of UDA will contribute to the project by leading Workpackage 2 and by essentially collaborating with Workpackage 3 which require knowledge in the areas of modeling, metamodeling, model transformation as well as model difference.

2.4.4 Caixa Mágica Software



Organization description:

Caixa Mágica Software (CMS) is the SME company responsible for the development and market of the Linux Caixa Mágica, the portuguese Linux distribution. Linux Caixa Mágica is today present in hundreds of portuguese computer stores and specialized resellers, Hospitals, Municipal Authorities, 1.100 schools spread all over the country, more than 10 universities and thousands of enterprises. CMS is involved in the deployment of more than 100 CMS appliance firewalls in portuguese army units and a large scale migration of 20.000 computers for the Ministry of Justice since 2006. CMS was one of the partners of the EDOS project, an FP6 Strep, having participated actively in the different workpackages.

Scientific competencies and specific skills for this project:

Caixa Mágica Software has a small but focused research team. Their policy is to develop and be at the state-of-the-art in three well-defined topics of research and be among the world top-class researchers. One of their chosen topics is meta-installers [TDL⁺07]. Since 2006 they define new features for Apt-get, the most used meta-installer in the world. Their team developed the first rollback approach and this feature is being integrated in apt-rpm and contributed to Apt main branch (Debian).

Contribution to the project:

Caixa Magica's skills in meta-installers together with their know-how in managing Linux Caixa Mágica repositories and the previous research work can provide a strong industrial set of competencies to MANCOOSI and strong leadership of WP3.

2.4.5 INESC-ID

**Organization description:**

INESC-ID is a not-for-profit privately owned institution dedicated to advanced research and development in the domains of electronics, telecommunications and information technologies. INESC-ID was awarded the status of “Laboratório Associado” by the Ministry of Science, Technology and Higher Education in recognition of the high scientific merit of the institution according to external international and independent evaluations. INESC-ID is mainly owned by Instituto Superior Técnico (IST), the School of Engineering of the Technical University of Lisbon. Most of its PhD researchers are teaching staff of IST. It integrates more than 70 PhDs and 100 post-graduation students.

Scientific competencies and specific skills for this project:

INESC-ID has been involved in Boolean Satisfiability (SAT) and Pseudo-Boolean (PB) Solving and Optimization since 1995 with the development of GRASP by João Marques-Silva. The most competitive SAT solvers available today (zChaff, BerkMin, MiniSAT) are based on the concepts of non-chronological backtracking and clause learning introduced in GRASP. In the last decade, members of INESC-ID have further developed new competitive SAT/PB solvers and proposed several new techniques to improve its performance. Current state-of-the-art SAT/PB solvers are extremely fast due to the use of techniques to prune the search space and dedicated data structures. SAT/PB solvers have been recently applied to upgradeability problems.

Contribution to the project:

Members of INESC-ID have experience in developing and applying SAT/PB solvers to real world problems. They have already participated in the SAT competition and the PB evaluation, and are involved in the organisation of the PB evaluation from its first edition. INESC-ID planned contribution includes the study and development of specialized upgradeability solvers, to be applied not only to metadata but also to real data, as well as supporting the organization of the international competition of constraint solvers for upgradeability problems.

2.4.6 University of Nice-Sophia Antipolis (JRU - Joint Research Units)



Organization description:

With a staff of more than a thousand of researchers and teachers, the university of Nice-Sophia Antipolis develops researches in most of the scientific fields and is committed to research excellence in mathematics, physics and computer sciences to name a few.

Scientific competencies and specific skills for this project:

The CeP project (which stands for “Constraints and Proofs”) belongs to the I3S laboratory, a CNRS and university of Nice Sophia Antipolis joined research laboratory (UMR 6070). The CeP project gathers constraint programming experts and proof technique experts.

The CeP project has a significant knowledge in the handling of continuous constraints, and in the use of constraints for the verification and validation of software. Its main contributions range from global optimisation to dedicated solvers, global constraints and combination of different solving techniques like SAT and CSP over finite domains.

Contribution to the project:

The CeP project members will actively contribute to the project by working essentially on workpackage 4 through their expertise in constraint programming, optimisation and logic programming. They will also be involved in the workpackage 5 and take part in the definition of CUDE.

2.4.7 Tel Aviv University



Organization description:

Tel Aviv University is the largest university in Israel, offering an extensive range of study programs and research in the arts and sciences, within its Faculties of Engineering, Exact Sciences, Life Sciences, Medicine, Humanities, Law, Social Sciences, Arts and Management. The School of Computer Science maintains a leading edge in both teaching and research. Extensive international collaboration between faculty members and colleagues abroad, and cooperation with Israel high-tech sector, advance and enrich the School wide-ranging research program.

Scientific competencies and specific skills for this project:

The Databases Research Group, led by Prof. Tova Milo, is a world leading group in the Databases and Web-based information management field. The group research areas include advanced data base applications such as data integration, XML and semi-structured information, web-based and peer to peer applications management and monitoring. The group research focuses on studying the fundamental problems that are raised by modern information and knowledge management systems and determining novel solutions to solve these problems. Over the last three years the group had specialized in the management of information related to Open Source Software, in the context of the EDOS EU project.⁵ The group expertise in data management as well as in open source software management are essential to the present project.

Contribution to the project:

The group contribution is most significant in WP2 and WP3 of the project. Specifically, major tasks in WP2 require the modeling and querying of software data, which is precisely the group expertise. Similarly, WP3 involves the management of updates and the monitoring and analyzing of streams of data (software changes/updates in this case), which again is one of the group expertise. The Tel Aviv research group specializes in the management of information distributed over the Web which is one of the the challenges to be tackled given the distributed nature of open source software managements. The group had already previously collaborated successfully with various members of the project (most notably Edge-IT, Caixa Magica and Paris Diderot). This collaborative work is expected to continue here.

⁵<http://www.edos-project.org>

2.4.8 ILOG SA



Organization description:

ILOG delivers software and services that empower customers to make better decisions faster and manage change and complexity. Over 2,500 corporations and more than 465 leading software vendors rely on ILOG's market-leading business rule management systems (BRMS), supply chain planning and scheduling applications as well as its optimization and visualization software components, to achieve dramatic returns on investment, create market-defining products and services, and sharpen their competitive edge. ILOG was founded in 1987 and employs more than 800 people worldwide.

Scientific competencies and specific skills for this project:

ILOG is a worldwide leader in optimization technologies and provides high-quality industrial optimization components, ILOG CPLEX being the most widely known. This leadership has been achieved thanks to more than twenty PhDs working in optimization R&D who are highly visible in the academic world, publishing papers or organizing workshops and conferences.

Contribution to the project:

ILOG has strong experience in solving combinatorial problems in industry, like transportation and manufacturing, and has also some experience in product configuration with two products dedicated to these problems (Configurator and JConfigurator). Nevertheless, software configuration is quite different, and the Mancoosi project is a good opportunity for ILOG to learn in this specific domain, technically, but also in terms of market in the *software* industry.

ILOG will mainly contribute to the workpackage WP4 by exploring and providing algorithms for solving the combinatorial aspects of the problem. ILOG is interested in developing a multicriteria optimization model for the upgradeability problem, and in developing a solver for this problem. We expect to reuse part of this solver in other application areas. For example, we think that the stability of solution is an important criterion for the upgradeability problem, and, thus, we will need to work on stability of solution. As stability is a common criterion in many applications and in a wide range of domain (scheduling, time-tabling,...) we expect to design some generic solutions and to reuse them in other contexts.

ILOG will also be involved in the workpackage WP5 for defining the CUDF.

2.4.9 University of Louvain



Organization description:

The Catholic University of Louvain, founded in 1425, was split in 1971, giving birth to two independent entities, UCL (Université catholique de Louvain) and its sister university KUL (Katholieke Universiteit Leuven). While the KUL remained in Louvain, UCL established itself in the new town of Louvain-la-Neuve. UCL currently has more than 20000 students (including 4000 foreign students), 3000 teaching and research personnel, and 1800 technical and administrative personnel. The Department of Computing Science and Engineering at UCL (INGI) is part of the Faculty of Applied Science. The department is recognized worldwide for its contributions to dependable distributed computing, networking, constraint and logic programming, programming languages, and software engineering. Major releases by the department include the KAOS methodology and its tools for requirements engineering and the open source Mozart Programming System.

Scientific competencies and specific skills for this project:

The Programming Languages group at INGI is recognized for its contributions in programming languages, distributed programming, computer science education, open source software development, and constraint programming. The group is currently the main developer and maintainer of the Mozart Programming System, a major open source platform for program development, distributed programming, programming education, and constraint programming (see www.mozart-oz.org). The group is currently coordinating the European project SELFMAN, which brings together two open source consortia, Mozart and the ObjectWeb consortium, to develop techniques for self management of large-scale distributed systems (see www.ist-selfman.org).

Contribution to the project:

UCL will participate in the development of a tool providing transactional updates over a loosely-coupled distributed system, together with the other participants of WP3. This tool will draw on the expertise of UCL in building loosely-coupled distributed systems with imperfect failure detection (such as the Internet). UCL will develop a solver for resolving upgradability constraints and integrate it with the transactional tool, together with the participants of WP4. The specification of the upgradability process will be based on the model developed in WP1; UCL will participate there to ensure that this specification is appropriate for the transactional tool and its constraint solver.

2.4.10 Pixart Argentina



Organization description:

Pixart is the company responsible for the development and marketing of Rxart Linux, the Spanish Linux distribution. Linux Rxart is today present in hundreds of Latin American computer stores and specialized resellers. It is used daily in many hospitals, municipal authorities, schools, universities and the Army in Argentina, as well as Chile and Peru, and is gaining acceptance in larger markets.

Scientific competencies and specific skills for this project:

Pixart has a small but focused research team. Our policy is to develop and be at the state-of-the-art in a few well-defined research and development areas, ranging from hardware development to meta-installers. Our team developed the first Rxart installer, based on the Debian infrastructure, so we have working experience with the problems in the apt-get installer, and its integration in the Rxart product.

Contribution to the project:

Pixart will bring to the consortium the Rxart Linux know-how in managing Linux Rxart repositories, and platform configuration; it will also provide a valuable testbed for the tools developed in the project.

Chapter 3

Impact

3.1 Strategic impact

Complex systems like Linux distributions and other large Open Source Software (F/OSS) projects (Apache, Eclipse, OpenOffice.org...) rely on a software distribution model where new elements can be added to the system or existent ones can be updated. Today, with the growing number of F/OSS bundles and the complex web of constraints at stake, this model is facing new challenges in the area of upgradeability.

The negative impact of an unsatisfactory trustworthiness in the management of the complexity of the Open Source infrastructure goes far beyond the F/OSS industry. A reliable management of the F/OSS infrastructure is critical for the different European industries to be independent of the powerful American software houses.

At the same time, user expectations of “just working” systems, ignoring the growing complexity and inter-dependencies of computer systems, have never been higher.

Institutions and enterprises are starting out research in this direction. The EDOS project¹ funded within the FP6 was a pioneer, and its outcomes are being carried out by others. Opium, a Univ. of California [TSJL07] project in collaboration with Linspire company, is one of the groups that followed the steps of EDOS and are now publishing and deploying tools on the subject.

The Mancoosi scientific challenges can hardly be tackled by a single European institution since different research areas are necessary and since there is the need for a strong liaison between research centers, universities and industry. Beside, F/OSS success requires the involvement of a strong and wide international community.

The Mancoosi consortium has reached an interesting balance with different kinds of organizations, from six different countries. Universities (University of Paris Diderot, Nice, L’Aquila, Tel Aviv and Louvain), Research Centers (INESC-ID), a worldwide leading company (Ilog) and SME Linux distribution companies (Edge-IT, Caixa Mágica and Pixart), which constitute the Consortium, have the knowledge and the conditions to reach Mancoosi goals and therefore to have a strong impact on research and industry.

The Mancoosi consortium proposes to provide a set of algorithms, techniques and tools for managing the complexity of the F/OSS infrastructure. WP2 modeling outcomes will allow to

¹<http://www.edos-project.org>

formalize the upgradeability problem. WP3 transactional upgrades will provide algorithms and software tools which allow robust roll-back and distributed retrieval of packages with positive impact on F/OSS software installation but also on the distribution of hardware drivers. The study of solvers in WP4 and WP5 will result in better algorithms and will pursue the improvement of the state of the art in SAT and PBO problems. This will have impact not only on the dependency solving topic but also in other NP-complete problems.

These specific points will result in a “higher level of software reliability” and “improved mastering of complex systems” defined as expected impacts of the Objective ICT-2007.1.2 of the Work Programme 2007. In fact, much of the drawbacks mentioned by organizations in deploying Linux relate to the difficulty in installing or upgrading software and lack of hardware support, topics that are directly or indirectly addressed by the present project.

Mancoosi also expects to reach other impact of Objective ICT-2007.1.2, namely the creation of “new opportunities for SMEs” and other companies through open and standard platforms for the “next generation operating systems”. Without the proposed enhancement of software distribution, it will be very difficult to foster this new generation of operating systems. They will simply miss the distribution of components and will fail in its adoption.

Collaboration to international standards is planned to be done not only individually by partners (such as the case of Edge-IT/Mandriva and Caixa Mágica conformance with LSB and other Linux Foundation standardization efforts) but also by the Consortium. The CUDF format designed within WP5 will be considered to be submitted to the relevant standardization organizations.

In modeling the software artefacts and the upgradability process, Mancoosi will have an impact on the ongoing work within the QualiPSo Integrating Project, running until 2010. Among other goals, QualiPSo aims indeed at “identifying factors for trustworthiness of F/OSS products”. Mancoosi will contribute to the identification, the modeling and the instrumentation of the trustworthiness of the upgrade process.

Linux distributions are the basis of more and more computer infrastructures. Their quality and reliability are then crucial to all systems based on Linux. Mancoosi outcomes will be integrated in distributions and other projects for the following reasons:

- it is package type neutral (both deb and rpm),
- it is clearly integrated in open source and commercially related activities,
- it is open source itself,
- it will help companies to integrate their offer for open source.

Progress on automatic tools for upgradeability problems will have an important impact to the administration of software package *repositories* done by software distribution editors. The problem consists here in upgrading the packages comprising a whole repository of software packages. This is a leap in complexity since we do not just have to maintain the consistency of the relations between packages, as it is in the case for the upgradeability problem as seen by the administrator of a single client machine. Rather, the invariant that we have to maintain between versions of the repositories is now expressed by a quantification over all possible installations that can be performed based on a version of the repository. In most cases this means ensuring that *every* set of packages that could be installed relative to the old version of the repository can also be installed in the new version.

This is a very real problem that currently has to be solved every day in case of the Debian *testing* distribution which evolves by upgrades of its packages. This is currently done by hand with some rudimentary tool support. With more than 20.000 available packages it is almost impossible to find optimal solutions by hand, and major bulks of package migrations, like those induced by upgrades of fundamental libraries, can currently only be achieved by brute force and at the price of breaking many package relations. The urgent need of good automatic tool support for this task has recently been recognized by the participants of the Debian Quality Assurance meeting which was held in December 2006 on the premises of Linex in Badajoz, Spain. We are confident that the techniques we propose to develop in this project will be used in the Debian context.

Finally, it is important to remark that the technology developed in Mancoosi will provide a strategic advantage for various European enterprises: Edge-IT and CaixaMagica will be able to provide for their user base a much more manageable infrastructure, ILOG will profit from the advanced techniques to improve its component model, and third party software vendors (even proprietary ones) will have a common reference model for integrating their products in the various Linux distributions.

Mancoosi is going to significantly change the way complex software systems are managed today, and become a reference for the software systems of the future.

3.2 Plan for the use and dissemination of foreground

Open Source. The consortium as a whole is entirely committed to the Open Source distribution model. As such, all the software deliverables as well as the intermediate revisions will be freely accessible to the public. Moreover, a homepage on the internet will be dedicated to the project, with links and/or software to download, including reports and articles.

By nature. By nature, the project aims at developing a set of tools that will address fundamental stability and manageability issues for a wide set of software platforms, ranging from servers in the data centers to the very desktop or laptop computers used by millions of people in day to day work. These tools will disseminate the consortium technologies worldwide, and highly increase the visibility of European R&D on the planet.

Beyond the lifetime of this project, the source code will live the life of regular Open Source software, with volunteers that may pick up the job and carry on. Since the tools we will develop have high visibility, due to the millions of actual users that will adopt them, there is no doubt that the work will be adopted by the community.

Industrial exploitation

The processes tackled by Mancoosi relate to core engineering and business activities of the three Linux software editors Edge-IT, Caixa Mágica and Pixart taking part in the project. The Mancoosi achievements will hence have direct repercussions on the user experience with the Mandriva Linux / Caixa Mágica operating systems. The industrial exploitation of Mancoosi outcome will consist in embedding the Mancoosi tools into the products of the two editors. The tool chain produced by Mancoosi will lay the foundations for a new generation of solutions able to address with a greater efficiency the upgrade management of enterprise workstations and servers, which currently represents a critical issue in terms of resources and risks. We intend to stress the following industrial applications within the project :

WP2 - Models for the description of software artifacts and the upgradeability pro-

cess

Even if this metamodel is crucial to prepare the rollback function, this task also has a direct industrial application : improving support productivity and quality. When receiving a support demand, the first step is always to determine the state and the history of the problem. We propose to give the possibility to the user to send the machine status description and the associated history generated by the tools developed in WP2 to the support team. With these informations, the support team will be able to solve the problem more easily.

The possibility to improve the support quality and to increase its productivity, is crucial for the industrial partners as it is a key revenue for open source industry. This tool is useful for Linux distribution editors but also for software editor support operations.

At the same time, we will use this possibility to add a "system history" within the Edge-IT next generation technical asset management tool (Pulse 2).

WP3 - Rollback feature

The software set up rollback feature is one of the most common demand for Linux distributions.

Edge-IT will integrate it in its next generation technical asset management tool (Pulse 2). Pulse integrates today a possibility to deploy a machine and then keeps its software profile updated compared to a reference profile. Once prepared at the packaging level, the possibility to roll back a whole update on one or many machine will be added to Pulse, in its second version (Pulse 2).

Rollback features will be included in future versions of the different distributions. In order to test this idea, a limited "server distribution" will be designed. The marketing argument for this distribution will be "roll back included". The whole distribution will be built around this feature, and it will be guaranteed that during upgrades, it will be possible to roll back to a previous software state.

The market share of the different distributions in the project is small. We expect this feature to boost significantly this market share, and the associated revenues.

WP4 and WP5 - Upgradeability solvers This technology offers the possibility to adapt an existing distribution to different uses :

- Embedded market: small footprint is preferred;
- Corporate market: reversibility of upgrades is preferred;
- Consumer market: functionality is preferred to reversibility.

Additional industrial exploitations

Outcomes from Mancoosi will also provide new models of deploying a Linux Distribution. With better dependency solving meta-installers, Edge-IT and Caixa Mágica will have larger software repositories with several versions of the same package. This is important for appliances and embedded Linux where older versions are used for long periods.

Caixa Mágica plans to differentiate from other Linux Distributions with a stable and powerful meta-installer based on Apt and with integration of algorithms devised in Mancoosi. This technical argument will be used to achieve better results in our product at stores and on-line sites. Cross-selling with professional services and training is also expected. Finally, our line of security appliances will integrate also the meta-installer and therefore contributing to the stability of the overall solution.

ILOG has strong experience in solving combinatorial problems in industry, like transportation and manufacturing, and has also some experience in product configuration with two products dedicated to these problems (Configurator and JConfigurator). Nevertheless, software configuration is quite different, and the Mancoosi project is a good opportunity for ILOG to learn in this specific domain, technically, but also in term of market in the *software* industry. Moreover, ILOG plans to reuse part of the Mancoosi solver in other application areas, when several criteria need to be optimized, or when stability of solution is needed like for the upgradeability problem (scheduling, time-tabling,...).

Hype. The open source community relies heavily on hype, be it good or bad. But it does have interesting effects: the source code is opened to the scrutiny of everyone, from different age, background, experience, or technical preferences. Only interesting software will be successful with this crowd, for no other criteria than usability, interest, and sometimes source readability if the code is not maintained anymore. Hype goes up easily if something addresses a common desire, it even goes higher if there are actually something to try on, and it is at its highest if one can freely download it. We plan on advertising our software through those channels as the development process reaches maturity, using well-known website such as freshmeat.net, sourceforge, and slashdot. Our measure of success will be the feedback we get from the Open Source community, be it remarks, ideas, or even source code.

Public Bodies. Researchers at the public bodies from the consortium will publish their results via their standard activities, like journal or conference articles.

Bibliography

- [Ecl05] Eclipse. Eclipse Modeling Framework (EMF), 2005. <http://www.eclipse.org/emf>.
- [Mar07] Claude Marché. Termination competition. <http://www.lri.fr/~marche/termination-competition/>, 2003–2007.
- [MBDC⁺06] Fabio Mancinelli, Jaap Boender, Roberto Di Cosmo, Jerome Vouillon, Berke Durak, Xavier Leroy, and Ralf Treinen. Managing the complexity of large free and open source package-based software distributions. In *ASE*, pages 199–208. IEEE Computer Society, 2006.
- [MZ07] Claude Marché and Hans Zantema. The termination competition. In Franz Baader, editor, *Proceedings of the 18th International Conference on Rewriting Techniques and Applications (RTA '07)*, Lecture Notes in Computer Science, pages 303–313, Paris, France, June 2007. Springer Verlag.
- [OMG03] OMG. *Meta Object Facility (MOF) 2.0 Core Specification, OMG Document ptc/03-10-04*. <http://www.omg.org/docs/ptc/03-10-04.pdf>, 2003.
- [PSS02] F.J. Pelletier, G. Sutcliffe, and C.B. Suttner. The Development of CASC. *AI Communications*, 15(2-3):79–90, 2002.
- [SAT] SAT. The international SAT competitions web page. <http://www.satcompetition.org/>.
- [SS06] G. Sutcliffe and C. Suttner. The State of CASC. *AI Communications*, 19(1):35–48, 2006.
- [Sut07] George Sutcliffe. The cade atp system competition. <http://www.cs.miami.edu/~tptp/CASC/>, 2007.
- [TDL⁺07] Paulo Trezentos, Roberto DiCosmo, Laurière Laurière, Mario Morgado, João Abecasis, Fabio Mancinelli, and Arlindo Oliveira. New Generation of Linux Meta-installers. *Research Track of FOSDEM 2007*, 2007.
- [TSJL07] Chris Tucker, David Shuffelton, Ranjit Jhala, and Sorin Lerner. Opium: Optimal package install/uninstall manager. *Proceedings of the 29th ACM/IEEE International Conference on Software Engineering (ICSE)*, IEEE Computer Society Press, 2007.