Project no: 004337

CALIBRE - Co-ordination Action for Libre Software Engineering for Open Development Platforms for Software and Services

Coordination Action

Priority IST-2002-2.3.2.3 - Open development platforms for software and services

Deliverable D4.5
Libre Software Educational Programme Specification

Due date of deliverable: April 30th 2006
Actual submission date: May 31th 2006

Start date of project: 1st June 2004 Duration: 24 months.

Organisation name of lead contractor for this deliverable:
Universidad Rey Juan Carlos, Spain

Revision 1.4

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract

Education and training on libre (free, open source) software is an open problem for all actors interested in the availability of well prepared, professional, knowledgeable human resources who can understand and take reasonable decisions in the domain of libre software (including its use, development, maintenance, marketing, promotion, etc.).

Taking into account those interests, many learning programs on libre software have flourished during the past years. However, most of them deal with the first need that has been identified: proficiency in libre software tools and technologies. This approach, being focused on the specific details of those tools and technologies, is quite valuable in order to train experts in the corresponding domain, but is not enough to provide the holistic background that many institutions demand today.

In this report, we address a scenario which could be considered as the second generation in libre software training: the compendium of knowledge and experiences needed to deal with the many facets of the libre software phenomenon. For this goal, we have considered higher education as the best possible framework. In the current context of the European Space of Higher Education (ESHE), this new discipline matches perfectly the definition of a postgraduate (master) study. In this report we propose the main guidelines of a program on libre software of that kind.

In summary, the studies designed in this report are aimed at providing students with the knowledge and expertise that would make them expert in libre software. The programme provides capabilities and enhances skills to the point that students can deal with problems ranging from the legal or economic areas to the more technically oriented ones. It does not (intentionally) focus on a set of technologies, but approaches the libre software phenomenon from an holistic point of view. However, it is also designed to provide practical and real world knowledge. It could be offered jointly by several universities across Europe, within the framework of the ESHE, or adapted to the specific needs of a single one. In addition, it could also be adapted for non-formal training.

Document history

Version	Status	Date	Author(s)
1.4	Feedback from MoLOS, CALIBRATION, etc	May 31st	jgb
1.3	Full report reviewed and completed	May 18th	jgb
1.2	Curriculum section updated	April 27th 2006	herraiz
1.1	Modifications and update after comments	April 18th 2006	herraiz
1.0	First complete version	February 28th 2006	jgb

List of authors:

- jgb: Jesús M. González Barahona
- herraiz: Israel Herráiz

Contents

Chapter 1

Introduction

Libre software[1] is becoming not only an interesting area within the software industry, but is also attracting more and more interest by researchers. This interest is focusing in many different aspects of libre software, from the business models that are being experimented to the development models in use by currently successful projects (which is the specific target of this report), to the legal implications of using and distributing certain programs.

In this context, there is an increasing need of IT professionals with good knowledge of the libre software phenomenon, so that they can understand the implications of libre software in all kinds of organizations (companies, public administrations, etc.), and in the society in general. In this report we offer the design of a postgraduate programme that addresses this need, providing students with the capabilities and skills needed, from a holistic and multifaced point of view. The programme is not (intentionally) focused on training on specific technologies which happen to be implemented with libre software. Although it also deals with them, the main focus is on the general knowledge and expertise needed to deal with libre software as an integral phenomenon, as will be shown in the next section.

1.1 Libre software as a matter of study

Libre software is becoming a field of study and expertise in itself. From a technological point of view, there are libre software solutions in almost any field of the information technology landscape. During the last years, the quantity of libre software available has grown exponentially, according to any parameter we may consider[2]. This trend is expected to continue during the next years, as the interest by companies, public administrations and other actors is increasingly more evident, not only as users, but also as producers and improvers of libre software. Therefore, technologies implemented with libre software (in some cases as a choice between other implementations, but in some others as the only option) are clearly of interest to all actors in the software industry.

However, these technologies are only the most visible aspect of libre software. Some other aspects, more deeply hidden, are at least equally important, and maybe more, to understand the

[1] In this paper we will use the term "libre software" to refer to software which fits in the definition of "free software" of the Free Software Foundation or in the definition of "open source software" of the Open Source Foundation.

[2] For instance, the number of lines of libre software is doubling every 18 or 24 months, and similar growths can be observed on the number of developers, the size of its market, etc.

libre software phenomenon. Among them, legal, economic, sociological and management issues are prominent.

From a legal point of view, libre software licensing, and the implications of software patents are arcane (for IT professionals) but highly important issues. The implications of choosing the right license to distribute a program, or the conditions that have to be met when using a software under a certain license are quite important in the libre software environment. The relationship between the patent portfolio of a company (and all the software patents in existence, in those jurisdictions where they are valid) and the libre software used or produced by that company have also to be managed with great attention. Specific legislative provisions dealing with libre software in public administrations are also becoming of importance in some areas.

With respect to economy, business models are of course an active and quite important area. Companies and other producers of libre software are interested in evaluating the resources needed to ensure the sustainability of the production of a certain system, and the long-term availability of sources of income, when the software is produced for profit. But many other aspects are also interesting, such as how to build communities of developers and users (including companies, research centers, relevant users, and even volunteer developers) in a way that all actors are benefited, how to build business models based on providing services, etc. All of them are quite interesting to define the strategies of companies with respect to libre software.

Sociological aspects are also relevant, specially those related to voluntary development, to leadership of libre software communities, to attraction of human resources and external contributions, and of course, to motivation (of special interest in most libre software projects, driven by voluntary work). Many differences with traditional software development can be tracked to sociological aspects, and in particular the mix of hired and volunteer developers raise interesting issues that are important to address.

Project management, and in general, management relationships within projects are also quite important. Since libre software projects usually follow development models quite different from those common in the industry, managing a libre software project is quite different from managing a traditional software project. In addition, a certain culture of management and coordination, and even tools supporting it, have developed over the years in the libre software community. Knowing those practices and tools is quite important to successfully join or lead a libre software project.

Some other aspects are also important (among them, of course, specific technologies). But these are probably the most different and important ones (when comparing with the usual environment in traditional software development).

In all these fields, it seems clear that IT engineers should get a deeper knowledge of libre software. The postgraduate programme presented in this report is precisely targeted at guaranteeing that well trained professionals are produced, and that they are able to deal with all the scenarios related to libre software.

1.2 Main skills and capabilities

In summary, the main target of this report is to present a postgraduate programme with the aim of promoting in the students the skills and capabilities usually present in an IT engineer who has been involved in an active libre software environment during several years, facing the real-world problems related to it. This profile includes several different faces, which correspond to the usual aspects of problems in libre software, and is specially targeted at project managers, decision makers, technology prospectors, and similar positions in companies, public administrations and other organizations with a potential interest in libre software.

Later in this report the specific skills and capabilities promoted by this programme will be presented in detail. Meanwhile, they can be briefly summarized in the following (non exhaustive) list:

- Familiarity with the usual licenses, and licensing models in libre software projects, and capabilities of reasoning about their impact and consequences.
- Knowledge about the business models being tested in the libre software environment, and the strategic and economic consequences for companies and other organizations of being producers or users of libre software.
- Ability of reasoning about the social and personal implications of libre software. This includes the design and ability of discuss public policies on the matter, but also the understanding of motivation of agents involved in the production and promotion of libre software (be them individuals or organizations, be them volunteer or for profit).
- Comprehension of the usual development, management and coordination models in libre software development, and their implications for companies or other institutions interacting with them.
- Expertise on the main tools and procedures used in libre software development, distribution and use, including those used in libre software communities.
- Capabilities on reasoning about the suitability of a given libre software solution, comparing between solutions of similar technical characteristics, and comparing between proprietary and libre software solutions.
- In general, sufficient level of knowledge about the history of libre software, and its main milestones, facts and projects, to be able of reasoning about its future, and its implications in other areas.

In addition, it is expected that students attending this programme can also acquire capabilities related to specific technologies and solutions. These matters would be open to different implementations by different degree-granting institutions, and will depend on the needs identified in the market, on the preferences of the students, and on the expertise available to the institutions. These capabilities would empower the students to be active, from a technical point of view, in those fields in which libre software has already shown its potential.

Summarizing, this programme will train IT engineers to deal with libre software both as a new kind of technical solution, but also as way to use, develop and build technologies using new paradigms.

1.3 Main guidelines of the programme

To address the targets already mentioned, and to provide students with the specified capabilities, the master programme is structured in three blocks, each consisting of 30 ECTS credits (usually, one semester)[3], thus being of a total of 90 credits. Adaptations of the programme are also provided for studies of 60 and 120 credits, and specialization studies of 30 credits are also possible. All these possibilities will be described later in this report.

The three blocks can be summarized as follows:

[3]By definition, one academic year is considered to be equivalent to 60 ECTS credits. ECTS credits include all the effort devoted by a student to the subject (not only teaching hours).

- Core block. It will include subjects which are considered as basic for the understanding of the libre software phenomenon, and will cover different approaches and points of view. The block will be open by an introductory subject, which will provide the skeleton on which the rest will lie. Then, we propose subjects on legal issues, economic aspects and business models, sociologic and ethic aspects, which should provide the "non-technical" background needed. In parallel, subjects on development environments and tools and libre software management will provide exposure to more "technical" issues. After following these subjects, students are expected to have acquired knowledge and expertise about libre software from the most important points of view, and be able of reasoning and taking decisions about it.

- Specialization block. This block will include subjects aimed at providing the students with specialized knowledge and capabilities. The exact subjects will depend on the institutions delivering the degree, the expertise available to them, their perception of the demands of the students and the actors in the libre software world, etc.

- Practicum. Will provide students with practical experience and hands-on involvement with real problems and issues related to libre software. The practicum will include a master's thesis and practices in companies, public administrations, libre software projects, or similar.

These blocks, targeted to IT engineers, could be completed with other subjects for students with other backgrounds. Although it is not the main goal of this report, the programme could be retargeted to other professionals (with backgrounds on law, business, or others), complementing it with some basic technological subjects, and providing a suitable specialization block.

1.4 Brief history and antecedents of this report

This report is based in part on the work performed by the MoLOS[4] (Master on Libre / Open Source Software) working group during several years. This group started in 2002, promoted by five European Universities, and funded by the Regional Government of Madrid. The first ideas about a postgraduate programme on libre software, and the first designs that have lead to this report, can be tracked to the developments by that group.

The MoLOS working group was open from the beginning, and several other Universities and experts from several countries joined since 2003. The proposed programme was evolving, as feedback from more parties was obtained, and more experiences were taken into account. Several public presentations of the programme (usually in combination with meetings of the MoLOS group) took place during those years in several places of Europe: Madrid (Spain, Nov. 2003), Brussels (Belgium, Dec. 2003), Soissons (France, Dec. 2003), Madrid (Spain, Feb. 2004), Barcelona (Spain, Jan. 2005), Genoa (Italy, Jul. 2005), Madrid (Spain, May 2006).

Since 2004, the MoLOS group joined the efforts by the CALIBRE Coordinanted Action with respect to the definition of a postgraduate programme suitable for implementation in any country of the European Space of Higher Education. The MoLOS working group has also established relationships with SIGOSSE (a project funded by the European Commission, focused on libre software and education) and TOSSAD (also funded by the European Commission), both interested in the topic of education about libre software.

Currently, MoLOS incudes experts from more than 20 higher education institutions all over Europe, and several other persons from companies and other institutions (mainly from Europe, but including several people from outside Europe).

[4] `http://www.nongnu.org/masterlibre/`

This report has further ellaborated the proposals by the MoLOS group, and has detailed and put into writting several other contributions, either from members of the group, or from other parties. The first results of the work done in the CALIBRE project were presented in the MoLOS group meeting in Barcelona, in January 2005, when it was decided to join the efforts of both MoLOS and CALIBRE with respect to education on libre software.

The first public presentation of the proposed postgraduate programme after this joining of efforts was at the Educational Symposium held at the Open Source Software Conference[5], in July 2005 in Genoa, Italy, in which feedback from several third parties was obtained.

A more complete version of this report, very close to the final one, was presented in the CALIBRE workshop held in Madrid, on May 5th 2006. Since most of the attendants came from companies with interest on libre software, the feedback obtained in it was mainly related to industrial concerns. This feedback was the main input before the final version of the report.

[5] http://oss2005.case.unibz.it/

Chapter 2

Formative goals and profile of competences

The main formative goals of the proposed programme, and an overview of the profile of competences that the students enrolling it would acquire were already introduced in section 1.2. In this chapter, they are revisited and detailed.

2.1 Formative goals

In summary, the main formative goal of this programme is to complement the capabilities and knowledge of persons with a background on IT technologies, so that they can deal with the problems and specific issues related to the adoption, production, promotion, and study of libre software, and the implications of mixed environments with libre and proprietary software. This main goal can be refined in the following list of more detailed objectives:

- To allow students to understand the libre software phenomenon, its history, the motivations of its main actors, and its main implications for the whole IT field (and specially for companies and public administrations adopting or producing it).
- To provide students with a general background on the regulations affecting the software industry, including copyright and software patents issues, with a specific aim at explaining how they affect libre software.
- To provide the students with detailed information about licensing issues related to libre software, and practical experience on how problems have been managed in this area by companies, public administrations and other organizations (including libre software projects).
- To explain the different legislative and regulatory measures, specifically related to libre software, introduced in several parts of the world, and the main actions of public administrations that promote, regulate or make it difficult the adoption of libre software in their jurisdictions.
- To present to the students the business models related to libre software, discussing their implications, and their relationship with traditional business models in the software industry.

- To provide students with knowledge and methodologies that empowers them to analyze the economic impact of libre software in organizations (either as producers or consumers of software).
- To allow students to understand issues related to sustainability of libre software projects, both from an economic and a sociologic point of view.
- To show students the inner working of libre software communities, the relationships of individuals and companies and of volunteers and hired professionals working in them.
- To let students understand the specialties of libre software development, including the differences between different projects, but also the aspects which are common to most of them. In addition, to let them also understand the similarities and differences with more traditional approaches to software development.
- To explain students the mechanisms by which libre software projects deal with quality, reliability, security and maintainability aspects, and their relationship with more traditional approaches to software development.
- To make students capable of leading a libre software project, and of understanding the issues related to human resources management in those projects.
- To provide students with expertise and first-hand experience in the use of tools and systems commonly used by libre software projects.
- To give students first-hand experience with motivation, meritocracy, trans-cultural collaboration and other sociologic aspects usually important in libre software development.
- To explain the main implications of libre software for public administrations, and non-profit organizations.
- To summarize the current landscape of libre software solutions, their main characteristics, their suitability to conform to user needs, and how they compare with their proprietary counterparts.

It is important to notice that knowledge and capabilities on specific technologies implemented with libre software is not a main goal of this programme. However, that is of course a good complement, and is in fact addressed by the current proposal, mainly in its specialization and practicum blocks. In this respect, some complementary goals can also be identified:

- To immerse students in specific libre software projects, letting them understand in detail their inner working, their culture, their development processes and the software they produce.
- To provide students with detailed information about the libre software solutions available in a give field, their advantages and problems, how they compare to each other and to their proprietary software counterparts, etc.
- To explain in detail the strategies and business models of certain companies related to libre software, for instance those in a give sector, showing also the impact on that sector and on the competitors using more traditional business models.

- To provide students with complementary knowledge and capabilities in development tools and systems common in libre software. For instance, students can learn programming languages common in libre software projects, or integrated development environments in common use in them.

A given implementation of the programme presented in this proposal would usually consider only one of these complementary goals.

2.2 Profile of competences

If the objectives mentioned in the previous section are reached, students successfully completing the programme would have the following provide of competences:

- Management profile
  - Be able of comparing different libre software systems from legal, economic, and technical points of view, to decide the one best fitting a specific need, being able of evaluating its impact.
  - Be able of designing and managing the deployment of, or migration to, libre software solutions, given the constraints of a certain organization.
  - To be able to define a strategy related to libre software by a company, public administration, or other kind of organization.
- Project leader profile
  - To be able to lead a libre software community, promoting it, and letting it grow and sustain over time.
  - To be able to manage global software development groups producing libre software, including those with mixed (volunteer / hired) human resources.
- Development profile
  - To be able to collaborate in most libre software projects, and of using proficiently the tools they use for coordination and project management.
  - To be able to browse the libre software projects in existence, and identify in them systems and pieces of code of interest for a given development team, and of knowing how to act in that case (e.g., deciding on whether it is convenient to join a given project to improve development rate).

It should be noted that, since the programme is for a postgraduate course, and students are expected to enroll in it with a graduate degree of an IT profile, all these competences are in addition to those they already had obtained.

In addition, specific implementations of the programme could provide students with other complementary profiles of competences, thanks to the specialization and practicum blocks. For instance, some of the following technical profiles could be provided:

- To be able to design and administering computer networks and software systems based on libre software.
- To be able to secure networks and systems (based or not on libre software) using libre software tools.

- To adopt state-of-the-art development practices and technologies found in libre software projects for in-house software development.

Chapter 3

Structure of the programme

The postgraduate program is structured in three blocks of 30 ECTS credits each, with a total of 90 credits, or one and a half academic years. The three blocks are:

- Core block: basic common ground for all the students.
- Specialization block: specialized education to students, who will usually choose from a list of optional modules.
- Practicum: real, hands-on knowledge, of some specific aspects of the libre software phenomenon.

Other structures can be derived from this basic design to meet special needs (such as specific regulations in some countries, specific students' backgrounds, needs of non-formal education programs, etc.)

3.1 Admission and graduation requirements

The basic proposed programme is oriented towards graduate students with a background on information technologies (computer science, computer engineering, software engineering, telecommunications engineering, etc.). However, students coming from some other scientific and engineering backgrounds could be admitted in the programme in two circumstances:

- Having them pass an accession test in which they may show that the have enough knowledge and expertise (maybe acquired by professional experience) to follow the programme.
- Having them follow an additional (and previous) block which provides them with enough knowledge and expertise on information technologies to get benefit from the rest of the programme. This block could be (depending on the specific backgrounds) of 30 credits.

In order to successfully complete the programme, students should successfully complete all the subjects in the core block (30 credits), at least 30 credits corresponding to subjects in the specialization block, and the activities in the Practicum (30 additional credits). The programme could usually be completed in one year and a half, although specific implementations could maybe allow students to complete it in just one year (performing Practicum activities in parallel with the specialization block), or in two years (spreading the Practicum activities over one whole year, part time, for instance to make it more easy for students to maintain a job).

Subjects in the specialization block could have enrolling restrictions, such as having specific backgrounds (e.g. proficiency in C and C++ programming) or being grouped (such as enrolling System Administration only if the student also enrolls Network Administration).

Implementation of programmes derived from the basic version could have specific admission and graduation requirements.

3.2 Core block

This block is aimed at providing a complete panoramic of libre software from the more relevant points of view. After following it, students should have knowledge and competences similar to those of an expert in libre software who has learnt through experience during several years. An emphasis on factual, proven data and accepted knowledge would be fundamental for the success of the academic design of these subjects.

The subjects in this block are described in the following subsections.

3.2.1 Introduction to libre software

Effort: 4 credits

Objectives: To provide a common background to all students enrolled in the master. To provide an introduction to the subjects which will come later.

Contents:

1. Introduction
2. What is libre software
3. Legal issues and their implications
4. Ethics of libre software
5. Economy of libre software
6. Business models
7. Development models and libre software engineering
8. Overview of the libre software landscape

3.2.2 Legal and licensing issues

Effort: 4 credits

Objective: To address legal and licensing issues related to libre software. Students will be prepared to understand the implications of libre software licensing, of choosing a license, of using a software under a given license, etc. In addition, they will also be confronted to regulations affecting libre software in some jurisdictions, and to the implications of licensing on business and development models, and in the strategies of companies.

Contents:

1. Introduction and motivation.
2. Basic background on regulations about software (copyright, software patents, etc.)
3. Libre software as a legal concept

4. Copylefting and other important aspects of licensing
5. Main licenses (description)
6. Choosing a license (implications)
7. Specific regulations about libre software in some regions
8. Case studies

3.2.3 Economy and business models

Effort: 4 credits

Objective: To show economic models and business cases in the libre software world. This model is in several points different from the "traditional" one, and it has strong links with sociological issues, also to be covered within this master.

Contents:

1. Introduction.
2. The simple economics of Libre Software
3. Competition.
4. Business models and industrial organization.
5. Governance.
6. Incentive theory and motivations.

Weblinks:

- `http://www.firstmonday.dk/issues/issue3$\_$3/ghosh/`
- `http://www.people.hbs.edu/jlerner/simple.pdf`
- `http://web.mit.edu/evhippel/www/opensourceexecsummary.pdf`
- `http://web.mit.edu/evhippel/www/opensource.PDF`

Recommended books:

- DiBona, Christ; Sam Ockman and Mark Stone (eds.). *Open Sources. Voices from the Open Source Revolution.* O'Reilly, 1999
- Chris DiBona, Mark Stone, Danese Cooper (eds.), *Open Sources 2.0, The Continuing Evolution*, O'Reilly, 2005

3.2.4 Sociology and ethics

Effort: 4 credits

Objectives: To show the motivations for voluntary development, communities and ethics of the libre software movement.

Contents:

1. Introduction
2. Motivation and ethics

3. Libre software developers
4. Libre software communities
5. Social networks

Weblinks:

- http://www.flossproject.org/report/index.htm
- http://widi.berlios.de/paper/
- http://www.nd.edu/$\sim$oss/Papers/PositionPaperMadey.pdf
- http://www.fsf.org

Recommended books:

- DiBona, Christ; Sam Ockman and Mark Stone (eds.). *Open Sources. Voices from the Open Source Revolution.* O'Reilly, 1999
- Chris DiBona, Mark Stone, Danese Cooper (eds.), *Open Sources 2.0, The Continuing Evolution*, O'Reilly, 2005

3.2.5 Development environments and tools

Effort: 4 credits

Objectives: To give a background to students about the tools and environments common in libre software development.

Contents:

1. Introduction
2. Development environments
3. Programming languages and related tools
4. Source code management
5. Documentation and tools to generate documentation. Integration of source code and documentation
6. Development-oriented frameworks
7. Finding the right components. Repositories and searches in the Open Source development model

Weblinks:

- http://curso-sobre.berlios.de/introsobre/
- http://cvsbook.red-bean.com/
- http://www.loria.fr/$\sim$molli/cvs/doc/cvs$_$toc.html
- http://www.sourceforge.net
- http://savannah.gnu.org
- http://www.berlios.de
- http://www.faqs.org/docs/artu/documentationchapter.html

Recommended books:

- Per Cedervist. *Version Management with CVS.* Signum Support AB.
- Oram and Loukides. *Programming with GNU Software.* O'Reilly.
- Norman Walsh.*DocBook: The Definitive Guide.* O'Reilly.

3.2.6 Project management

Effort: 4 credits

Objectives: Common practices and issues related to libre software project management

Contents:

1. Introduction
2. Collaboration mechanisms and tools
3. Leadership
4. Follow-up of libre software projects
5. Building and coordination of libre software communities
6. Project management in mixed environments (volunteer and hired developers)
7. Main activities: Bug tracking, versioning, configuration management, etc.

Weblinks:

- `http://www.firstmonday.dk/issues/issue4$\_$12/bezroukov/`
- `http://linfe.it/AICA-OpenSource/localdocs/OSS-giugno03/Papers/Iannacci.pdf`

Recommended books:

- Feller and Fitzgerald. *Understanding Open Source Software Development.* Addison-Wesley.
- Torvalds and Diamond. *Just for Fun: The Story of an Accidental Revolutionary.* HarperBusiness.
- DiBona, Christ; Sam Ockman and Mark Stone (eds.). *Open Sources. Voices from the Open Source Revolution.* O'Reilly, 1999
- Chris DiBona, Mark Stone, Danese Cooper (eds.), *Open Sources 2.0, The Continuing Evolution*, O'Reilly, 2005
- Eric Raymond. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* O'Reilly.
- Donald K. Rosenberg. *Open Source. The Unauthorized White Papers.* IDG.

3.2.7 Case studies

Effort: 6 credits

Objective: Broad and comprehensive study of libre software projects, companies, organizations, etc.

Contents (to be updated according to latest developments):

1. Introduction
2. Linux
3. The BSD family
4. GNOME and KDE
5. Apache
6. Mozilla
7. OpenOffice.org
8. ObjectWeb
9. Eclipse
10. Debian, Ubuntu, Red Hat, Mandriva and other distributions

Weblinks:

- `http://curso-sobre.berlios.de/introsobre/`
- `http://www.kernel.org`
- `http://www.freebsd.org`
- `http://www.openbsd.org`
- `http://www.netbsd.org`
- `http://www.gnome.org`
- `http://www.apache.org`
- `http://www.mozilla.org`
- `http://www.openoffice.org`
- `http://www.debian.org`
- `http://www.redhat.com`
- `http://ag.cs.uvic.ca/$\sim$dmg/research/papers/dmg$\_$wosse2002add.pdf`

3.3 Specialization block

This block is composed of an open set of subjects, which should be different, and possibly change in time, according to the interests detected by the institutions offering the programme, to the availability of experts in the mattes involved, to the changes in technology, etc. This block should permit students to gain specialized knowledge in some specific aspects related to libre software, or in technologies in which libre software is relevant.

Some possible subjects (all of them focused on libre software in the related areas) are:

3.3.1 Languages common in libre software projects

Effort: 4 credits

Objective: Different languages are used to make up building blocks of any software system. In this module we expect to make the student, who is supposed to have basic knowledge of programming, acquainted with the basics of the languages more frequently used in Libre Software projects in order to qualify him/her to, e.g., decide the best one for every application.

Contents:

1. Introduction.
2. Shell scripting.
3. Scripting languages (Perl, Python, PHP).
4. Compiled languages (C/C++/Java/C
5. What to use? Strengths, weaknesses and interfaces.

Weblinks:

- `http://libresoft.dat.escet.urjc.es/debian-counting/`
- `http://www.pyhton.org`
- `http://www.php.net`
- `http://www.perl.com`

Recommended books:

- Oram and Loukides. *Programming with GNU Software*. O'Reilly.

3.3.2 Version control and configuration management

Effort: 4 credits

Objective: The student should be aware of additional problems found in a decentralized, albeit cooperative, environment, and of the tools which can be used to overcome them.

Contents:

1. Introduction.
2. The versioning problem and their solutions: snapshots, differences, versions; branching and merging.
3. Concurrent Version Systems: CVS, Subversion, Arch.
4. Configuration management.

Weblinks:

- `http://linas.org/linux/cmvc.html`
- `http://www.daveeaton.com/scm/CMTools.html`

Recommended books:

- Per Cedervist. *Version Management with CVS*. Signum Support AB.
- Karl Fogel, Moshe Bar. *Open Source Development with CVS*. Paraglyph Press. Also available from `http://cvsbook.red-bean.com/`.
- Oram and Loukides.*Programming with GNU Software*. O'Reilly.

3.3.3 Deployment and administration of common libre software platforms

Effort: 4 credits

Objective: The student should understand and assimilate how libre software is commonly installed and configured. Administration tasks regarding the local system and network will also be shown.

Contents:

1. Introduction and history.
2. Installation of a GNU/Linux distribution.
3. Package management.
4. Kernel compilation.
5. File systems.
6. Software configuration.
7. System administration.
8. Network administration.
9. Other operating systems: FreeBSD.

Recommended books:

- Bill McCarthy. *Learning Debian GNU/Linux*. O'Reilly.
- Olaf Kirch and Terry Dawson. *Linux Network Administrator's Guide*. O'Reilly.

3.3.4 Security in libre software environments

Effort: 4 credits

Objective: To introduce students into common security aspects and fundamentals. To show the threats and tools that exist.

Contents:

1. Introduction.
2. Access control.
3. Malicious code.
4. Secure system planning and administration.
5. The orange book.
6. Cryptography.
7. Communication and security in networks.
8. Internet security.
9. Security tools.

Weblinks:

- `http://www.criptored.upm.es/planes/p$\_$si1$\_$uncoma.htm`

Recommended books:

- Richard Blum. *Open Source E-mail Security*. SAMS.
- Stuart McClure. *Hacking Exposed: Network Security Secrets and Solutions*. McGraw-Hill.

3.3.5 Libre software web based systems

Effort: 4 credits

Objective: To give students an ample idea of the different solutions that exist in the web-base libre software scenario.

Contents:

1. Introduction.
2. Web servers.
3. PHP-based systems (PostNuke, Claroline, etc.).
4. Perl-based systems (Slashcode, etc.).
5. Python-based systems (Zope, etc.).
6. Java-based systems (Jakarta/Tomcat, etc.).
7. Ruby on Rails.

Weblinks:

- `http://www.apache.org`
- `http://www.php.net`
- `http://www.postnuke.org`
- `http://www.slashcode.com`
- `http://www.zope.org`
- `http://jakarta.apache.org/tomcat/`
- `http://www.clarolineserver.com/urjc/EXPETSAI/`
- `http://www.rubyonrails.org`

Recommend books:

- Lauri and Laurie. *Apache: The Definitive Guide*. O'Reilly.
- Latteier and Pelletier. *The Zope Book*. SAMS.
- James Goodwill. *Apache Jakarta-Tomcat*. APress.
- Chromatic, Aker and Krieger. *Running Weblogs with Slash*. O'Reilly.
- Hughes and Zmiesvki. *PHP Developer's Cookbook*. SAMS.

3.3.6 Office tools

Effort: 4 credits

Objective: To provide the students knowledge about libre office suites and their development.

Contents:

1. Introduction.
2. Case study: OpenOffice.org.
3. Office development environment.
4. Designing Office applications.
5. Office Objects and object models.
6. Add-ins, templates, wizards and libraries.
7. Debugging and error handling.
8. Internationalization.
9. Document design based on their logical layout.
10. Open Document Format and related standards.

Weblinks:

- `http://www.openoffice.org`
- `http://development.openoffice.org/index.html`

Recommended books:

- J. David Eisenberg. *OpenOffice.org XML Essentials*. O'Reilly.

3.3.7 Multimedia

Effort: 4 credits

Objective: Introducing the student to the difficulties of the multimedia world: timing and real time issues, loss of precision, lossless and lossy compression, resource (bandwidth, memory, processor) management. Giving the student the ability to choose the best solution based on the initial system constraints.

Contents:

1. Introduction to multimedia computing.
2. Types of multimedia data and compression standards. Libre software compression tools and libraries for sound, image and video.
3. Timing, synchronization, QoS, resource management. Streaming software for reception and for broadcasting.
4. Authoring tools for static and dynamic images and audio.

Weblinks:

- `http://www.agnula.org/`

Recommended books:

- Ralf Steinmetz, Klara Nahrstedt. *Multimedia: Computing, Communications and Applications*. Prentice Hall.

3.3.8 Mathematic tools

Effort: 4 credits

Objective: To provide a common background to all students enrolled in the master about mathematical tools and suites that exist in the libre software world. This permits the student to evaluate them in order to choose the right tool library for a given target.

Contents:

1. Introduction.
2. Computer-aided mathematics and fields of application.
3. The R project and the R suite.
4. The S language.
5. PSPP, a free SPSS.
6. The BLAS, LINPACK and LAPACK linear algebra packages.
7. Graphs-generating software.

Weblinks:

- `http://www.r-project.org`
- `http://www.gnu.org/software/pspp/pspp.html`
- `http://www.netlib.org/blas/`

3.3.9 Database systems

Effort: 4 credits

Objective: This subject provides an initial background to all students enrolled in the master. It serves as well as an introduction to the elective subjects which will come in the second semester of the master.

Contents:

1. Introduction.
2. Information Systems and Databases.
3. The relational model and SQL language: a hands-in introduction.
4. Database core modules.
5. Developing in the database core.
6. Case studies: MySQL and PostgreSQL.
7. Developing database plugins.
8. Security in database systems.

Weblinks:

- `http://www.mysql.com`
- `http://www.postgres.org`

Recommended books:

- Bruce Momjian. *PostgreSQL: Introduction and Concepts.* Addison-Wesley:
- Kofler. *The Definitive Guide To MySQL.* APress.
- DuBois. *MySQL Cookbook.* O'Reilly.

3.3.10 Embedded systems

Effort: 4 credits

Objective: To provide a common background to all students enrolled in the master. To provide an introduction to the subjects which will come later.

Contents:

1. Introduction.
2. Requirements of embedded systems: size, speed, power.
3. Tailoring source for embedded systems.
4. Development: Architectures, emulators and toolchains.
5. Security, liveness and real time concerns.

Recommended books:

- DiBona, Christ; Sam Ockman and Mark Stone (ed.). *OpenSources. Voices from the Open Source Revolution.* O'Reilly.

3.4 Practicum block

The goal of this block is to expose students to real, first hand experiences of how libre software actually works in given environments. To make it happen, a part of the practicum block will be devoted to participation in real libre software projects, practices in companies deeply involved in libre software or in public administrations with experience on the matter, to perform detailed and in-depth studies of real cases related to libre software (libre software deployment, development, promotion, etc.).

The practicum block will be split in two different modules:

- Practice work.
- Master's thesis (20 credits).

3.4.1 Practice work

Effort: 10 credits

Objective: To provide the student with real experiences related to libre software, if possible from a multidisciplinary point of view.

Details: The student should perform her practical work under the direct supervision of an educator. The student will get detailed recommendations on how to get the most benefit from the experience. The student will write a report as a result of the experience, detailing the most important aspects of it.

3.4.2 Master's thesis

To fulfill all the requirements to obtain the master degree, students must write a original thesis, about some topic related to libre software.

Works based on experience of students in real libre software projects, or in experiences in the domain of libre software in companies, public administrations, research and other environments will be encouraged. The student can also build up on the experience acquired in the practice work module (see previous section).

The master's thesis does not need to be an original contribution, from a research point of view, but should be novel enough to be considered at least as a contribution to the body of knowledge describing libre software in different environments.

Effort: 20 credits

Objective: To let the student put into practice the knowledge and competences acquired during the programme, to analyze from an educated point of view a case related to libre software.

Chapter 4

Implementation, other options, general recommendations and perspectives

Depending on how many institutions participate in the master, on the availability of professors, on the target students, and on the acceptance of the program by students, we suggest several possible implementation scenarios (involving a single institution, or several institutions with a combination of distance and in-place learning, etc.), withut the goal of being exhaustive, and having into account that mixed scenarios would probably be also interesting.

In addition, some other options are also presented in this chapter, for instance combining the described studies with other fields, packaging them for other education levels (eg., graduate studies or non-formal education), etc.

The chapter ends with some general recommendations and perspectives for the future, related to the studies on libre software.

4.1 Implementation scenario: single institution

The single institution scenario is quite classic. A University offers the master program. Professors are mainly part of that University, with the possibility of visiting professors. Students enroll in that University, and attend the classes in the facilities of that University.

In this scenario, the master could be divided in three semesters. The first two of them, being part of the first academic year, and corresponding to the core and auxiliary subjects. The third semester would be devoted to the Practicum.

4.2 Implementation scenario: Intensive joint degree

In the case of several European Universities offering the master, an intensive and joint scenario would be possible. It has into account several limitations which are difficult to avoid in this scenario. First, professors have usually many difficulties in staying every year long periods of time (several months) outside their University. Second, students may have problems if they have to wander through Europe, moving to different Universities in different locations every some months.

To avoid these limitations while trying to maximize the educational impact of the master studies, we propose a schedule based on intensive periods where students and teachers get together in a given location, followed by longer stays, usually at their home University, combined with distance (but locally assisted) learning. The details of this proposal are as follows:

- Each subject will be taught in three periods, two of them in-place, the other as distance learning (and mainly devoted to personal work by the students). For a typical subject of 4 credits, about 30 hours of in-place, classroom teaching (including both lectures and practices) will be provided in two periods: about 18-24 hours at the beginning of the semester, and about 6-12 hours at the end of the semester.
- Twice a year (summer and winter), students will move to a given location (usually provided by one of the Universities participating in the program) to assist to in-place intensive classes during 4 to 8 weeks. The ending period of a semester will be followed by the starting period of the next one, at the same location. This simplifies travels and the logistics in general.
- The rest of the time students will be at their home Universities, or maybe in stays in other partner Universities (specially during the third semester). During these periods students will complete assigned personal or group projects (using groupware software to interact with other students). They will also use distance learning tools to complete their knowledge on the subjects, and will use videoconferencing and recorded sessions to assist to distance lectures and discussions. They will also have local advisors who will help them in their learning process.
- Students will have their semester exams right after the in-place period, when they come back to their home Universities, and when the semester has ended.
- During the in-place period, classes will be performed intensively (about 6-7 hours a day), combining practice classes, lectures, discussions, etc. In general, two subjects will be taught in parallel, one during the morning classes, the other during the afternoon ones.
- Optional subjects will be offered in second and third semesters, giving students the chance to select whichever they want (that is, not overlapping in schedule). During the in-place periods, students will attend 3-4 hours classes corresponding to optional subjects. This will be half the offer of each semester.
- The first part of the Practicum will be done during the second semester. The master's thesis during the third semester. During the in-place periods, students will arrange for prospective advisors of their master's work. The third semester will be specially well suited for students to visit other partner Universities. For instance, to be close to their supervisor on the master's thesis.

4.3 Other options

Implementing a master programme according to the design shown in chapter 3 is not the only option. Depending on the environment, the goals of the institution implementing the studies, and other parameters, many other options based on it are possible, both as formal and non-formal education. Some of these options are described in the following subsections.

4.3.1 Combinations with other fields of interest

There are at least two possibilities for combining the specialization in libre software with other fields of interest, without deep redesigning efforts:

- Use the specialization block (30 credits) to provide the student with the additional knowledge and capabilities in the other field of interest. If this is possible with those 30 credits, the programme will still provide a reasonable background on libre software, and will be completed with this other area, still maintaining an effort of 90 credits (usually lasting for 18 months). The practicum block should also be adapted to include expertise on the second field.
- Use an additional 30 credits for the core knowledge and expertise related to the second field. This way, the background on libre software is maintained, and a basic background on the other field can also be provided, at the expense of 30 more credits of effort (therefore, the programme would usually last for 24 months). The specialization block could include subjects related to both fields, when this were possible, and the same can be said with respect to the practicum block.

Some possible 'second' fields of interest which could be considered are: data communications, web services, advanced software development, database management and database applications, etc. All of them assume a background on IT, but in case the programme is adapted to other backgrounds (see section 4.3.2), it could also be combined with other fields, with a strong relationship to libre software, such as intellectual propriety law, economics of the IT sector, economics and sociology of innovation, etc.

4.3.2 Adaption to other backgrounds

The programme presented is based on the admission of students with a background on information technologies. However, it could be adapted to people with other backgrounds. This could be done in several ways (the list is not exhaustive):

- An additional block could be included at the beginning of the programme. That block would include all the basic IT-related knowledge and competences needed to be able to follow the next blocks. Given the contents of the core block, this will be mainly matters related to software development. Of course, the goal is not to try to have a software engineer in six months, but to empower students with enough capabilities to understand the technical IT-related subjects in the core block. The specialization block could be completed with subjects suitable to the background of the students (eg, in-depth legal matters related to libre software is the students enrolling have a background on law). The practicum block should also be adapted for these students.
- An additional block could be included at the beginning of the programme, as in the previous option, but removing the specialization block. That way, students could not enter into deep details, but could understand the most important aspects of libre software, maintaining their background, and spending only 90 credits on the task.

This master programme will provide students with specialized knowledge and competences in their background area, making it therefore possible for a student with a background in law to specialize in depth in the intellectual propriety and regulatory issues of libre software, or to a student with an economics background to specialize on economy of libre software and business models related to libre software.

4.3.3 Shortening the programme

The program, as proposed in chapter 3, could be shortened because of students needs or regulatory issues in the country of implementation. In the European Space of Higher Education, it is possible to have postgraduate studies of just 60 credits, and in some cases (e.g., when grade studies are long in duration), that maybe a good choice.

In our case, the most obvious way of shortening the programme is by removing the specialization block, which will let postgraduate students acquire knowledge and competences in libre software, to a certain depth, although they will not be able to specialize.

Another more complex option would be to redesign the practicum block, reducing it to maybe 15 credits, while maintaining a reduced specialization block of some other 15 credits.

4.3.4 Enlarging the programme

The program can also be enlarged, again when it is perceived by the implementing institution that that is convenient, or when national regulations recommend it.

The enlarging can be done in several ways. The most simple is by adding 30 credits to the specialization block, maybe organized in tracks. But probably what makes more sense is to complement the basic design with other fields of expertise, as was described in section 4.3.1.

4.3.5 Non-formal education

Although the basic programme presented in this report was designed with the idea of fitting in the recommendations of the European Space of Higher Education, other options are possible. One of them is particularly interesting, because of its great impact on professionals and organizations. It is the adaption of the programme to non-formal education. Although it is out of the scope of this report to deal with this issue in detail, in the following paragraphs we discuss some ideas about it.

The programme could, for instance, be offered to professionals interested in continuous training as a set of modules, which they could follow or not according to their interest and background. In this case, each subject would be the basis of a module, with the requirement that it should be designed in an as much self-contained way as possible.

This design would allow, for instance, professionals with a good general knowledge of libre software to skip the introductory module of the core block, while taking the rest in sequence. On the contrary, professionals knowing almost nothing about libre software, and with an interest in just being introduced to the matter, could follow that module (maybe deciding later to go on with some others).

Companies and public administrations interested in allowing their employees to be more familiarized with libre software could offer (probably as in-house training seminars) the modules they find more appropriate, according to the strategy of the organization, and the background and previous expertise of the employees.

In any case, since the subjects are already, as presented in this report, reasonably self-contained, the effort of adapting them to this continuous training scenario should not be much. For instance, it is likely that exactly the same materials used for subjects in a postgraduate programme could be used by the modules offered to professionals under this schema.

4.4 General recommendations

It is quite important that this programme is implemented according to the needs and expectations of both the employing organizations (companies, public administrations, etc.) and the professionals in the IT field. Therefore, it is quite convenient that each implementation of the programme maintain close contacts with professional organizations, with experts of recognized prestige in the field, and with companies and public administrations. Those parties should assess on the implementation of the master, and on its evolution. For this matter, it is recommended that a special body (an assessing committee, for instance) is established for each implementation of the programme.

In addition, the field of libre software, and the knowledge which the academic community is gaining about it, is changing very quickly. Therefore, it is also important that this programme is revised periodically (probably every two years is a good compromise between the adaption to new developments and the needed stability to allow for implementations to catch up). The feedback from institutions implementing it, from students, from experts in the libre software field, from professional organizations and from companies and public administrations is basic for these redesigns to adapt to the new needs that may arise in the future, and to correct defects of the current design.

Of course, implementations of the programme should also update their contents and even the design of some subjects as time passes, incorporating the new knowledge and experiences that for sure are going to reshape the world of libre software in the following years.

Another concern for implementations of the proposed programme will be the creation of pedagogical material. The strength of the libre software community will probably better show up through the collective and open creation and sharing of learning materials. Some actors have already taken this direction: pedagogical documentation is freely available for several curriculae, as it is the case for the master described in this report. But this step is still hard to get, as pedagogical documentation appears to be more tightly linked to the authors' personnality than it is the case for software or encyclopedian knowledge. It still has to be proved whether libre teaching documentation can become as successful as libre software has been.

4.5 Perspectives for the future

Within a few years, the spreading of the "libre" paradigm has got over several steps. At the begining, mainly hardcore developpers (often referenced to as "hackers") were concerned. Then came other computer scientists. Then IT managers. Now that libre software has become a "respectable" alternative for companies (whether using it or creating it), many other professionals are directly concerned by this libre paradigm: salesmen, lawyers, economists... The phenomenon has also reached other domains, since not only software, but also books, music, and many other artefacts can be built under libre licences (in fact, all that information technology allows us now to produce, copy, modify and transfer easily). Whether the libre phenomenon is really new or only a renewal of something old can be debated, but one thing is sure: very few is taught about it academic curriculae.

As the number and specialities of the persons concerned by the libre paradigm has increased, so did the need for new trainings. From an historic perspective, it is not surprising that the first trainings were focused on mastering tools and technologies used within libre software projects. Then came more holistic approaches. This is clearly the case for the master degree proposed in this report, focused on providing IT graduate students with knowledge and expertise in libre software.

Whereas this is very useful and necessary, it is only one step more in a long way. Further work is needed to advance in that way, since knowledge of the libre paradigm should not be restricted to the IT community. Many other professionals need to understand in detail the impact of libre software, and the libre phenomenon in general, in their professional context: licenses, author rights, business models, development models, specific assets and risks, collective creation, etc. Many of these aspects are relevant to non-IT professions. Matters explaining libre software and the libre paradigm should therefore be introduced into non-IT curriculae, not as a specialisation this time, but as a basic knowledge.

In fact, it is now clearly recognized that there is a need for students from any speciality to master many IT-skills. This observation has already led to the creation of IT-certificates, such as the European Computer Driving Licence[1], or, more recently in France, the C2i[2], aimed at undergraduate students. Among other IT-skills, knowledge about libre software issues should be tackled in those certificates. Whereas the ECDL is maybe too "low-level technical skills oriented" to reach such goal, certificates such as C2i are good candidates. Indeed, promoting and improving C2i-like contents with respect to the libre paradigm is an interesting perspective towards libre-awareness and e-citizenship.

[1] See for instance `http://www.ecdl.co.uk/`

[2] Certificat Informatique et Internet, `http://c2i.education.fr/`

Chapter 5

Current initiatives

In this chapter, the current situation of studies and educational programs on libre software will be summarized, by detailing and providing links to some representative cases already running.

5.1 Subjects and specific studies

There is many experience on subjects, mainly in PhD and master programs, dealing specifically with libre software or libre software development. In this section we mention, in no particular order, some of them (although for sure several other, also interesting and relevant, are missing).

5.1.1 Universidad Politécnica de Madrid and Universidad Rey Juan Carlos

Joaquin Seoane and Jesus M. Gonzalez-Barahona started in March 2002 a subject on libre software[1] in a PhD programme at Universidad Politécnica de Madrid (Spain). In 2003 it became a joint subject, also offered in a PhD programme at Universidad Rey Juan Carlos (Spain). The subject is targeted at presenting the libre software phenomenon from several points of view, including licensing issues, bussines models, economic impact, software development, project management, and some others. All the materials of this course (including a book, digital recordings of many classes, documents produced by the students, etc.) is available under libre licenses.

5.1.2 University of Victoria

Daniel German has been teaching a subject on open source software development[2] since 2004 at University of Victoria (Canada). Although it is focused on libre software development and project management, with a strong foundation in empirical software engineering, it also deals with other facets of libre software.

5.1.3 University of Göteborg

Mathias Klang and Jonas Öberg have been teaching the subjects "Free software/open source: philosophy and theory" and "Free software / open source: political economy" at University of

[1] `http://curso-sobre.berlios.de`
[2] `http://turingmachine.org/opensource/`

Göteborg[3](Sweden).

5.1.4 Technical University of Kaiserslautern

At the Technical University of Kaiserslautern, the subject "Open source practical course"[4], by Marcus Ciolkowski and Ralf Carbon, has been running since 2001 (therefore being one of the earliest in this area). It is targeted at undergratuate students.

5.2 Master programmes in Europe

5.2.1 Universidad Oberta de Catalunya

The Universidad Oberta de Catalunya[5] (Spain) started the first master programme on libre software[6], back in September 2003. This is a distance learning programme which has been evolving with time, and which will be offered, starting in 2006, as an official title within the European Space of Higher Education. The design of the master is currently very much in line with the recomendations in this report (although it is also different in several aspects). The experience of this programme, being the longest one of its kind, has been very valuable for the programme shown in this report.

Most of the materials offered to the students of this programme (including the reference books for many of the subjects, which have been produced specially for them) are available under libre licenses.

5.2.2 Università di Bologna

The University of Bologna (Italy) started in September 2004 a postgraduate program on technology of libre and open source software[7]. This programme has several resemblances with the proposal found in this report.

5.2.3 Università di Pisa

The University of Pisa (Italy) offers a master on the management of open source software[8]. It considers several backgrounds for students (ranging from law, economics, andn humanities to science). The curriculum includes technical subjects (such as programming, database and operating systems) and other subjects with a focus on law and economics.

5.2.4 Ecole Centrale de Paris

The Ecole Centrale de Paris (France) offers a "Mastère d'Ingénièrie des Systèmes Ouverts"[9] which has some resemblances to the programme proposed in this report, although it is much more oriented to teaching about technologies.

[3] http://www.gu.se
[4] http://wwwagse.informatik.uni-kl.de/teaching/osp/ws2004
[5] http://www.uoc.edu
[6] http://www.uoc.edu/masters/softwarelibre/
[7] http://www.almaweb.unibo.it/os_home.html
[8] http://www.master.netseven.it/
[9] http://www.msio.ecp.fr

5.2.5 Ecole Internationale des Sciences du Traitement de l'Information

The EISTI (France) open in 2005 a Mastère Spécialisé named "Open source entreprise et collectivité locales"[10]. A "mastère spécialisé" is a French curricula which comes after a master degree. This master program lasts for 15 months, split in one semester for lessons, three months for practice, and one further semester for a trainig period.

5.2.6 Programmes scheduled to open in 2006

- The Universidad de Extremadura (Spain) is scheduling a master on libre software to start in Autumn 2006.
- The "Master Ingénierie du Logiciel Libre"[11] is a brand new master degree scheduled to be open in Autumn 2006 by the Université du Littoral (France).

[10] http://www.mastereopensource.com/
[11] http://dpt-info.univ-littoral.fr/mediawiki/index.php/I2L:Accueil

Chapter 6

Acknowledgements

All the participants in the MoLOS grup deserve a special thanks. Specially, those drafting the first version of the MoLOS programme: Joaquín Seoane, Manuel Carro (Universidad Politécnica de Madrid, Spain), Jaime Villate (Universidade do Porto, Portugal), Francois Pelegrin (ENSEIRB, Bourdeaux, France) and Jesús M. González-Barahona (Universidad Rey Juan Carlos, Mostoles, Spain, also one of the authors of this report).

After that, several others have also contributed with many time, effort and ideas, in the context of the MoLOS working group. Just to mention some of them: Janis Gailis, Nils Ulltveit-Moe (Agder University College, Norway), David Megías (Universidad Oberta de Catalunya, Barcelona, Spain), Roberto di Cosmo (Universite Paris VII, France), Jean-Michel Dalle (UPMC, Paris, France), Eric Cousin (ENST, France), Anne Østergaard (DKUG, Denmark), Cornelia Boldyreff, Stephen Rank (Lincoln University, United Kingdom) Martin Michlmayr (Cambridge University, United Kingdom), Roberto Barbuti (University of Pisa, Italy), Giancarlo Succi, Barbara Russo (University of Bolzano, Italy), Renzo Davoli (University of Bologna, Italy), and Maciej Hapke (Poznan University, Poland).

The team at the Universidad Oberta de Catalunya have also been deeply involved in this effort, and their experience running their master on libre software (the first programme of this kind, to our knowledge) has been of great importance to us.

All the partners at the CALIBRE project have also been very supportive of this initiative, and their feedback (including that by the members of the CALIBRATION industrial forum) has been very valuable.

A big thank you also goes to team maintaining the Savannah libre software hosting site, which is providing infraestructure for the discussions and intiatives of the MoLOS group.

To finish, thanks to all individuals who helped to make the programme described in this report more useful: they are too many to mention, and some of them have even contributed anonymously.