Project no. FP6-028038

PALETTE

Pedagogically sustained Adaptive LEarning Through the exploitation of Tacit and Explicit knowledge

Instrument: Integrated Project

Thematic Priority: Technology-enhanced learning

D.MAN.12 – Updated version of PALETTE Open Source Strategy Report

Due date of deliverable: 30 September 2008
Actual submission date: 12 January 2008
Re submission: 31 May 2009

Start date of project: 1 February 2006 Duration: 36 months

Organisation name of lead contractor for this deliverable: GEIE ERCIM

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
P	Public	**PU**

Keyword List: PALETTE Services, Licenses, Open Source
Responsible Partner: Karen Marache, GEIE ERCIM

MODIFICATION CONTROL			
Version	Date	Status	Modifications made by
1.0	05-11-2008	Draft	Karen Marache
1.1	17-11-2008	Draft	Matthieu Farcot
1.2	01-12-2008	Draft	Karen Marache
1.3	17-12-2008	Draft	Matthieu Farcot
1.4	18-12-2008	Draft	Karen Marache
1.5	05-01-2009	Final	Matthieu Farcot
1.6	25-05-2009	Resubmission following rejection	Liliane ESNAULT

Deliverable manager

- Karen Marache, GEIE ERCIM

List of Contributors

- Karen Marache, GEIE ERCIM
- Matthieu Farcot, CRP-HT

List of evaluators

- Vincent Quint, INRIA
- Christine Vanoirbeek, EPFL

Summary

This deliverable is an updated version of D.MAN.08 “PALETTE Open Source Strategy” and presents some juridical and legal recommendations in terms of licensing, for the PALETTE services developed during the project. These recommendations are based (i) on the analysis of the questionnaire provided by the QualiPSo project and (ii) on the analysis of the audits performed by CRP-Henri Tudor on the Services with the FOSSology tool.

The outcomes of this deliverable will impact the final PALETTE Exploitation Plan (presented in deliverable D.DIS.12).

TABLE OF CONTENT

TABLE OF FIGURES

1 Introduction

The consortium of PALETTE has, since the beginning, decided to follow an Open Source licensing strategy for the various software services that were to be developed within the framework of the Project.
The project partners produced the wide range of PALETTE services that can now be listed, as the end of the project is in reach:

- Collaboration Services: CoPe_it!, eLogbook
- Knowledge Management Services: SweetWiki, BayFac,
- Information Services: Amaya, LimSee3, DocReuse

It is important to mention the fact that other pieces of software have been developed or used during the project but can solely be considered as Services for developers, and not for end users. Under such a category of tools dedicated to developers are for example software libraries, either as original creation or derivated from pre-existing code.

In the framework of our task dedicated to the Open Source Strategy, we focused on the Services dedicated to end users, in other words to the Services that may be used as such in real life after the end of the project, and that are not to be assimilated to just a brick upon which others might build an application.

These services are all expected to be exploited in a way or another at the end of the project, and each of their different exploitation opportunities will be detailed in a separate deliverable, D.DIS.12 “Final PALETTE Exploitation Plan”.

A study of exploitation opportunities can not be complete without an analysis of the software’s code as well as legal/juridical framework. This is highly understandable from a strict juridical security standpoint. As we shall see within this report, Free and Open Source software licensing can raise some difficult issues, mainly in the field of license interoperability.

During one year, both ERCIM and the CRP-Henri Tudor, with the help of the QualiPSo European funded project[1], have put in place and set up different documents and tools allowing an in depth licensing analysis of the PALETTE Services:

- In the first deliverable D.MAN.07 “Draft Open Source strategy”, we presented the first step of our analysis: the presentation of an exhaustive list of Open Source licences that could or should be used, and the results of the preliminary analysis conducted by INRIA Grenoble, based on a basic questionnaire submitted to the PALETTE development partners relating to the nature of the licensing and ownership strategies that each partner would follow considering the Services they have developed.

- In the second deliverable D.MAN.08 “PALETTE Open Source strategy”, we explained how the bottleneck in the flood of information we gathered with this preliminary questionnaire conducted us to collaborate with the persons involved in the “Legal issues activities” of QualiPSo European funded project. Working on an IPR

[1] QualiPSo Fact Sheet available at http://cordis.europa.eu/fetch?CALLER=PROJ_IST&ACTION=D&DOC=19&CAT=PROJ&QUERY=1187189361220&RCN=80465

tracking methodology, this project has helped us to determine if the services developed within the PALETTE project depended on free of protected know how, and by which means the licensing strategy for each Service would allow the set up of a sustainable Open Source exploitation strategy.

- In this last deliverable D.MAN.12 "Updated version of PALETTE Open Source strategy report", we first explain the methodology used in addition to the QualiPSo questionnaire for the licensing analysis, and then give the example of result we had on the Bayfac Service. Each Service's analysis would be taken into consideration in the PALETTE Exploitation Plan task. However, this last assessment task is not to be considered as a binding obligation. It was proposed to the PALETTE service developers as an optional check-up, designed to validate the information gathered from the QualiPSo questionnaires. This is for two obvious reasons. As we shall see, some licenses include technical parameters upon which a copyleft clause is activated or not. This technical clause implies that it is the way the code under license is implemented that will determine whether the developers are free to choose any other license or are bound in some ways by a specific licensing scheme. This implied that there was a need for a strong availability of the development team. On the other hand, the automated assessment based on FOSSology and the underlying treatment of the information incoming from it required that the code would be stable, and do not change anymore. This is indeed in some ways a need which is positioned at the exact opposite of what makes the real strength of free and open source software developments : community based developments require agility and the essence of the benefits that such a development scheme allows is precisely in the "release early, release often" paradigm. This implies that code changes in a real dynamic manner, following any needs that may arise. In such a situation, it would have been difficult to justify a stop in a continuous and value creating development partners that would have impacted negatively the quality of the PALETTE services developed.

2 Intellectual Property Rights tracking among PALETTE Services

As seen in the previous PALETTE reports on the topic, Free and Open Source Software (FOSS) licensing frameworks refer to specific means of managing the patrimonial life of digital creations protected by Intellectual Property (IP). Such licensing frameworks depend on the exclusive rights granted by IP to the legal owner of a creation. However, whereas IP rights compose what are called "negative rights", based on a philosophy of exclusivity and monopoly, the FOSS licensing frameworks turn this approach upside down to develop a global, open ecosystem focused on knowledge exchange. This open ecosystem does not rely on the usage of appropriability regimes that allow the capturing of sustainable competitive advantages on the software as such, but on a common, co-created platform to expand value-added opportunities of services and knowledge exchange.

"As advances in information technology and basic science transform the nature of innovation and increase the relative value of intellectual property, it is crucial that we modernize the systems for creating and protecting those assets. Only through a truly collaborative process can we design evolving systems that will foster the continuing progress that benefits all of us." (Source: Ronald Martin, University of Texas School Law, in Building a new ip marketplace report, IBM, 2006)

This cumulative process of derivative or larger works creation is a core element in explaining the success of Open Source projects developments. Open Source blurs the border between knowledge creators and knowledge users, and to be truly successful depends strongly on a community approach to software development. How such a system can work and remain sustainable depends strongly on the licensing scheme followed by the product upon which the community is build. This licensing scheme will organize the way IP is managed within the project. It will indicate, either in a broad or narrow sense, how rights granted by a licensor on a licensed development will impact the licensee. IP rights that can be included among the license are the following (see figure 1):

- Copyright, dealing with works of authorship such as artistic and literary creations;
- Industrial property, regrouping industrial patents and Trademarks, among others rights.

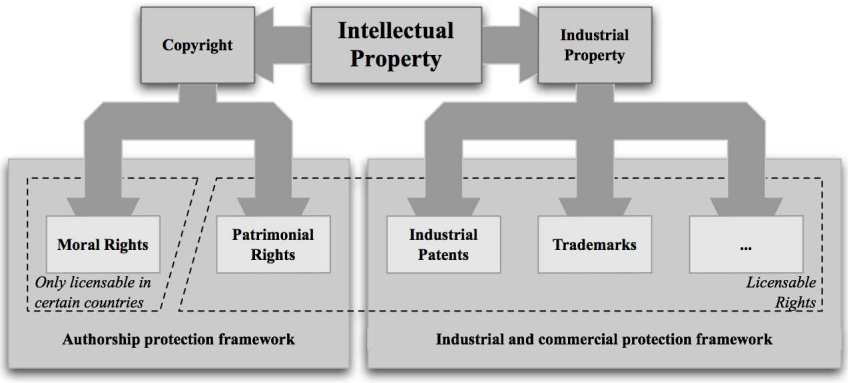

Figure 1 – An overview of licensable IP Rights

Before focusing on licenses “as such”, we shall introduce briefly the various and most common IP related rights implied usually in software creations. Software is usually created because of the existence of a specific, important or not, technical issue that we identify as a “technical problem”. This problem usually can be solved in various ways, which we identify as technical solutions. These solutions, should they respond to other criteria such as novelty and industrial applicability, are potentially patentable computer implemented inventions. The intellectual property assessment of the Services developed within the framework of the PALETTE Project therefore needs to include the patent dimension specific to industrial property. This is also the reason why a fair number of free and open source licenses contain specific clauses related to patent management.

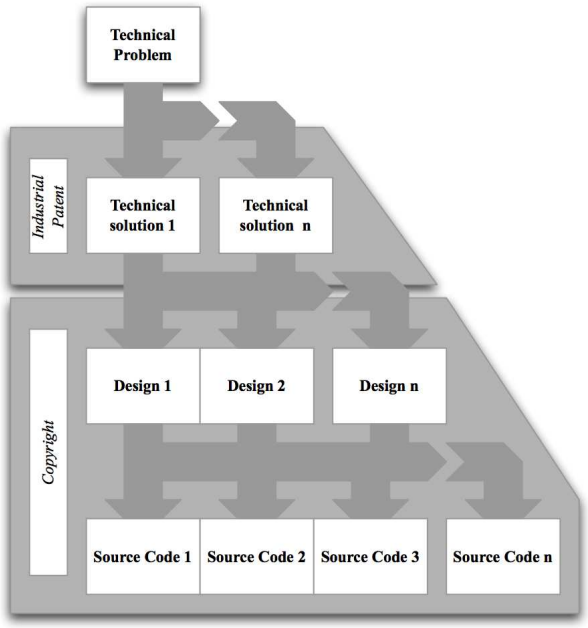

Figure 2 – IPR regimes in the case of software

As illustrated in figure 2, these solutions can be plural and heterogeneous in approach: the mean of designing a solution is not unique. In this situation, should the technical problems and needs be appropriately analyzed and transferred upon a specific design, they then can be organized upon the various steps needed to achieve the related technical solution. This technical design can then itself be transposed into a source code. These last two main steps are not protected by industrial patents, but by copyright.

So industrial patents and copyright do not protect the exact same thing, and the protection they grant can be combined. This explains why some FOSS licenses include specific regimes to organize not only the copyright related management issues, but also the industrial patent related ones. The same problematic might arise when a software is released to the public, under a protected TradeMark. Some Open Source licenses explain how each of these intellectual property assets all relating to the same product needs to be managed.

However, the organisation of rights and obligations among a license are potentially specific to a license, which means that the depth and breadth of granted rights and related obligations might not be compatible between two different Open Source licenses.

This raises the very technical issue of Open Source licenses interoperability.

As shown in figure 3, FOSS licenses depend on the principles of “some rights reserved”. But this approach can be plural, and this is the reason why three main sorts of FOSS licenses exist.

The most well known FOSS license, the Gnu General Public License, actually available under version 3, belongs to a group called “reciprocal licenses”. Such licenses impose a principle of Copyleft which relies on Copyright to ensure that any given software under said license cannot be captured into proprietary software. To do so, these license impose on the licensee an obligation to redistribute under the same license any derivative or larger work they create using the original code under reciprocal license. This strengthens the idea of a common pool, and imposes obligations on licensees for the sake of the protection of freedom.

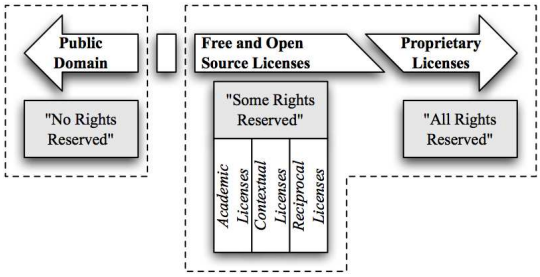

Figure 3 – FOSS licenses

Academic licenses, on the other hand, do not impose obligations on licensees apart from recognizing the paternity of the original work used. This implies that software under licenses composing this group, such as the BSD or MIT licenses, can be incorporated upon proprietary developments.

Finally, between academic and reciprocal licenses, a third set consists of contextual licenses. The latter depend on a technical context of usage to either trigger or not a copyleft obligation. The most well known example of such licenses is the GNU Lesser General Public License, or L-GPL, also now available under version 3.

As much different as they might seem, each of these approaches belong to Open Source licensing, as they all rely on the ten principles which are (source: http://www.opensource.org):

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of the Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product
The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral
No provision of the license may be predicated on any individual technology or style of interface.

The means of interpretation for these 10 principles are large, and explain why Open Source developments need IPR tracking management strategies.

3 QualiPSo methodology[2] impact on PALETTE

The objective of the collaboration with QualiPSo project was to easily audit the different services developed within PALETTE project, to know if their licensing constraints fit with the strategy and Exploitation Plan defined by the Consortium.

The QualiPSo IPRT[3] methodology consists of two distinctive phases organizing both an audit process and the related analysis outcome.
The initial audit process is based on a specific questionnaire (creating an initial declarative statement related to the development being audited), and relies secondly on the source code is processed through an automated code auditing tool designed to find any licensing related with an audit phase, to make sure the development process has produced a legal status which is compliant to the exploitation strategy of the Software, or which allows to perform corrective action(s) to modify the legal status accordingly.

[2] IPR Tracking: A methodology for Component Based and Collaboratively Developed software, in deliverable D1.4.1 "Report on the proposed IPR Tracking methodology" (final version to be released 1st quarter 2009), L. Grateau, M. Fitzgibbon, G. Rousseau, S. Dalmas (INRIA), QualiPSo project

[3] By Intellectual Property Rights Tracking (IPRT), we refer in this document to a set of process and actions aiming at defining the legal status of Software and monitoring its evolution during the development life cycle.

The Audit module is based on six steps or phases:
(1.) Development team provides a description of the software;
(2.) Goals and objectives of the audit are defined;
(3.) Legal status is determined by comparing "perceived" legal status – based on a questionnaire – and "determined" legal status – based on a code mining tool such as FOSSology™;
(4.) Problem Identification and Risk Evaluation is operated;
(5.) Critical problem solving is performed;
(6.) Residual Risk (if any) is covered by insurance before dissemination/distribution

In order to handle this Audit, the INRIA team involved in QualiPSo project built a questionnaire to collect related information (described in D.MAN.07).

The analysis of QualiPSo questionnaire raised different potential issues for the viability of the Exploitation plans the consortium could consider (first results of analysis presented in D.MAN.08).

For instance, regarding the first part of the questionnaire "Code and components", the analysis stressed the fact that we should pay particular attention to large number of external or modified components composing the Services, which could imply strong constraints on possible exploitation scheme.
Moreover, the second part of the questionnaire "Contractual context and Peripheral IPR", and in particular questions about "Global exploitation / dissemination scheme" alerted the consortium to the heterogeneity of licensing scheme for different services, that could be a problem if we want to provide an interoperable and extensible set of innovative services, for instance.

However, as the questionnaire was based on "declarative" answers, we could imagine that some important aspects or data could not have been provided.
This conducted us (being advised by the QualiPSo team and IPRT process) to envisage to back-up this questionnaire with an automatic treatment. The first automated audit has been used to BayFac software service, as shown in figure 4.

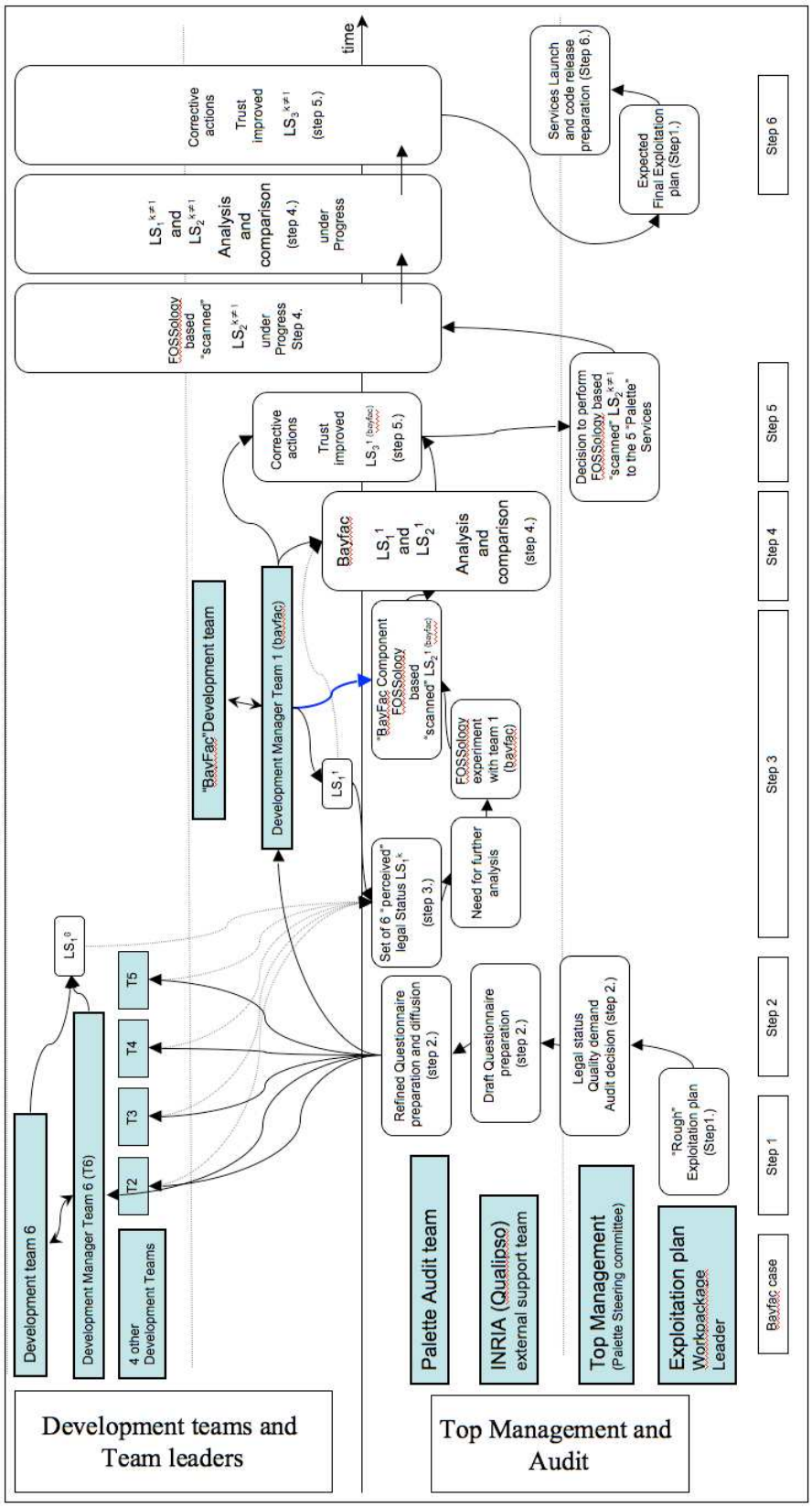

Figure 4 - PALETTE IPRT Process: example of BayFac

4 CRP-Henri Tudor methodology and use of FOSSology tool

Based on FOSSology (http://www.fossology.org), the treatment mines through of the various source codes to validate and allows for a complete legal risk evaluation. Auditing in such an automated manner the sources has allowed us to identify and propose solutions for all authorship-related, and compatibility issues between licenses that have arisen.

The audit team set up at CRP-Henri Tudor used first the full 6 steps IPRT methodology to the BayFac software service (developed within CRP-HT researchers' team). It leads to a clearly defined Legal Status for this software service in a quick process of only two iterations with the development team. Components redundancy, as well as an unused component presence in the release package, were identified and led to corrective actions and improvement of the release package quality.

Step 1: High level description of the PALETTE services

Open Source software corresponds to software licensed under an Open Source Initiative (OSI) certified license. The usage of third-party preexisting software within the developed PALETTE services makes of the later a derivative or larger work. Such derivative works, based on the inputs of others, are bound by legal obligations that might impact the licensing scheme of the service. In particular, licensing interoperability issues might appear therefore generating legal risks.

Hence, a high level description of the software (software architecture, functionalities, modules or components) is mandatory to allow actors to speak a common language and to have a functional representation of the software.

Step 2: Audit objectives

A first "Draft Exploitation Plan designed for PALETTE services" was released at the beginning of the project. In order to contribute to the "Final Exploitation Plan report", the Steering Committee proposed to work on "legal quality" of the software services developed within PALETTE, to know if their licensing constraints fit with this Exploitation Plan defined by the Consortium.

This audit should provide indicators in order to ensure the feasibility of the Exploitation Plan/ Business Plan for PALETTE services.

Step 3: Determination of "Perceived legal Status" for any PALETTE software and additional "FOSSology Scanned" Legal Status for BayFac Software

The INRIA team involved in QualiPSo project built a questionnaire to collect the related information.

The questionnaire is composed of two parts: "Code and components" and "Contractual context and Peripheral IPR". Hereafter are the results of the preliminary analysis of the questionnaire, provided end of December 2007.

First part: Code and Components

Almost all components are based on a large number of external components

- each external component must be clearly identified
- Some of them induce potentially strong constraints on possible exploitation scheme.
  - Identify carefully if the link between the external component and the tool induces contamination from GPL-like components
- Some of the components have been modified.
  - Open question about integration of modified components into release of PALETTE tools or contribution to original community (with leading question about maintenance of modified/forked version)

Second part: Contractual context and peripheral IPR

Contractual context

- Pre-existing know-how (PKH) have been identified
  - Check if the initial version of the PKH (before modification by PALETTE) is reachable.
  - Check if all IPR owners are member of the PALETTE consortium
  - Check how PKH must be interpreted according to the consortium agreement

Global exploitation / dissemination scheme

- Some of the PALETTE services already appear to have different dissemination strategy from GPL based to more permissive one based on MIT or LGPL licenses. For the case of GPL, the licensing scheme followed is copyleft, strengthening a common pool from which every derivative work will need to retain the same license as the original one. This is not the same situation for permissive licenses such as BSD or MIT, which leave the licensee the freedom to decide which license to choose for its derivative works.
- PALETTE project aims to provide an interoperable and extensible set of innovative services
  - Open question about heterogeneity of licensing scheme for different services
  - Depends on high level architecture implementing “interoperability” (ie what will be the nature of the link between the services)
- Global vs. local dissemination scheme from the point of view of PALETTE objectives,
  - Define objectives and priorities in terms of dissemination
  - Improve understanding at services level of the possible compatibility problem with global dissemination strategy
  - Identify blocking/critical point
  - According to available resources, roadmap, and priority, consider to solve some of the problem identified (for instance substituting a component under contaminating license), modifying the global interoperability scheme at the technical, legal or dissemination level

After this first analysis, a “automated legal status mining” of BayFac software has been performed. This work was done using the FOSSology license checker and lead to the Legal Status $LS_2^{1\ (Bay\ Fac)}$.

Step 4: Problem identification and Risk evaluation

This “automated legal status mining of BayFac” was compared to the $LS_1^{1\ (bay\ Fac)}$ “perceived” legal status of the development team and discussed with it.
No critical problem was identifying for BayFac software and risk to distribute it under a GNU GPL licence was considered to be very low.

An example of non critical problem detected by using FOSSology is the presence in the BayFac software archive used for the audit of a component not mentioned by the development team. After discussion it appears that this component was not used anymore.
The BayFac treatment under FOSSology is illustrated hereafter:

FOSSology is an automated licensing audit tool based on a metadata database built upon full and partial texts of numerous licenses. Using a web guided user interface front end based on Apache and PHP technologies, it allows the uploading of the source code of projects within a specific repository. The latter is then analyzed through an agent based treatment to find licensing patterns.

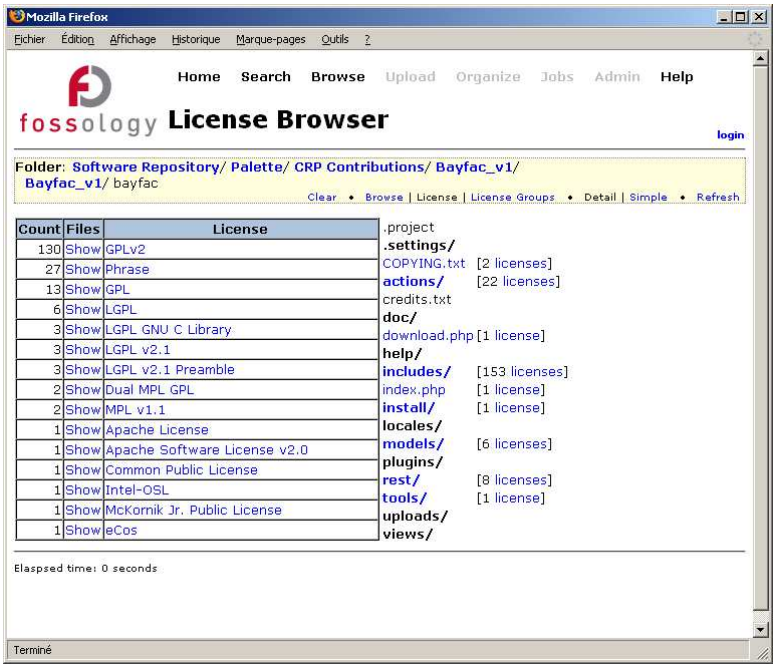

Figure 5 - The BayFac components license browser

Once the source code is uploaded, the various licenses of the components used are displayed as illustrated in the figure 5 above. This browser also allows navigating within the source code elements.

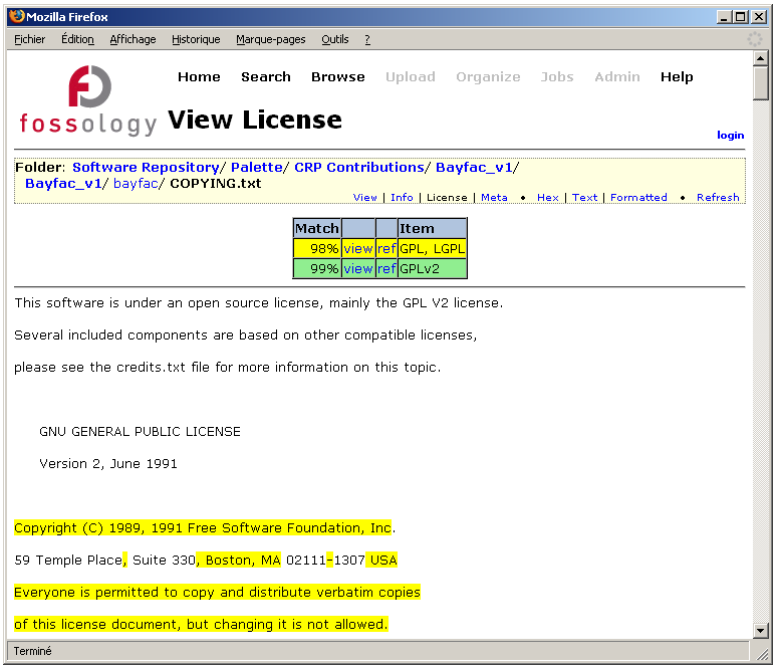

Figure 6 - Automated audit license matching analysis

License search underlines the various licensing legal references within the source code of the selected application, as illustrated in figure 6.

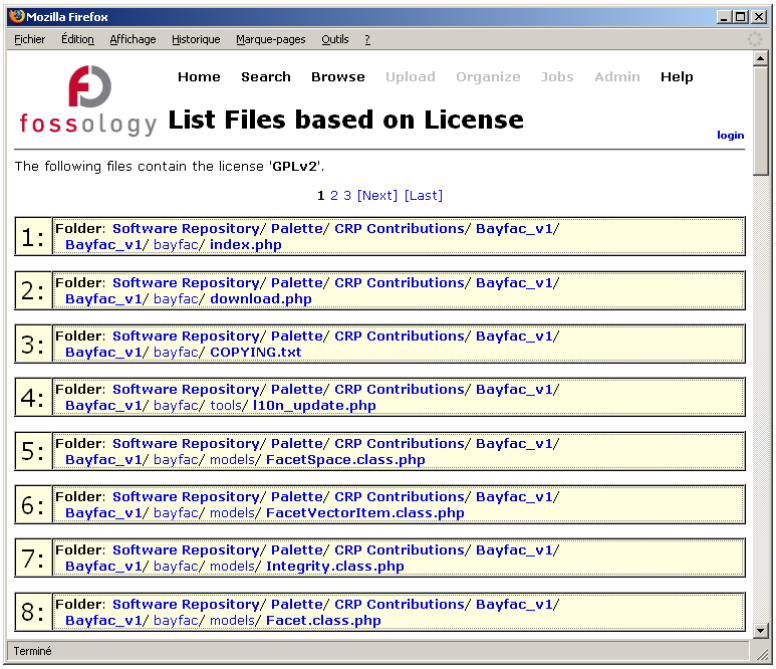

Figure 7 - License references finding

Finally, a global search can be done on all of the sources, showing the various components under license. In the case of the above figure 7, the search was done on components mentioning GPLv2 licenses.

FOSSology, as an automated auditing tool, creates a useful framework to find potential licensing issues. Raising these issues, it also allows for a global code quality analysis.

But although useful, FOSSology has serious limits. In particular, this tool only searches for licensing plain text paterns and references in a given source code. Should the references be removed, either voluntarily or not, FOSSology will be of no real use. Secondary, technical issues within FOSS licenses are difficult to tackle. Depending on the considered license, the mean of implementing the code under license will trigger or not a reciprocal, GPL-like, clause. This implies that the legal analysis should not solely be restrained to the usage of this tool. License identification is useful, but more information and treatment can be needed to know if issues appear or not.
However, beyond such limits, FOSSology remains extremely useful at a first code auditing level.

Step 5: Solve Blocking problems

After minor corrective actions, result $LS_3^{1\ (Bay\ Fac)}$ was obtained. It is presented figure 8.

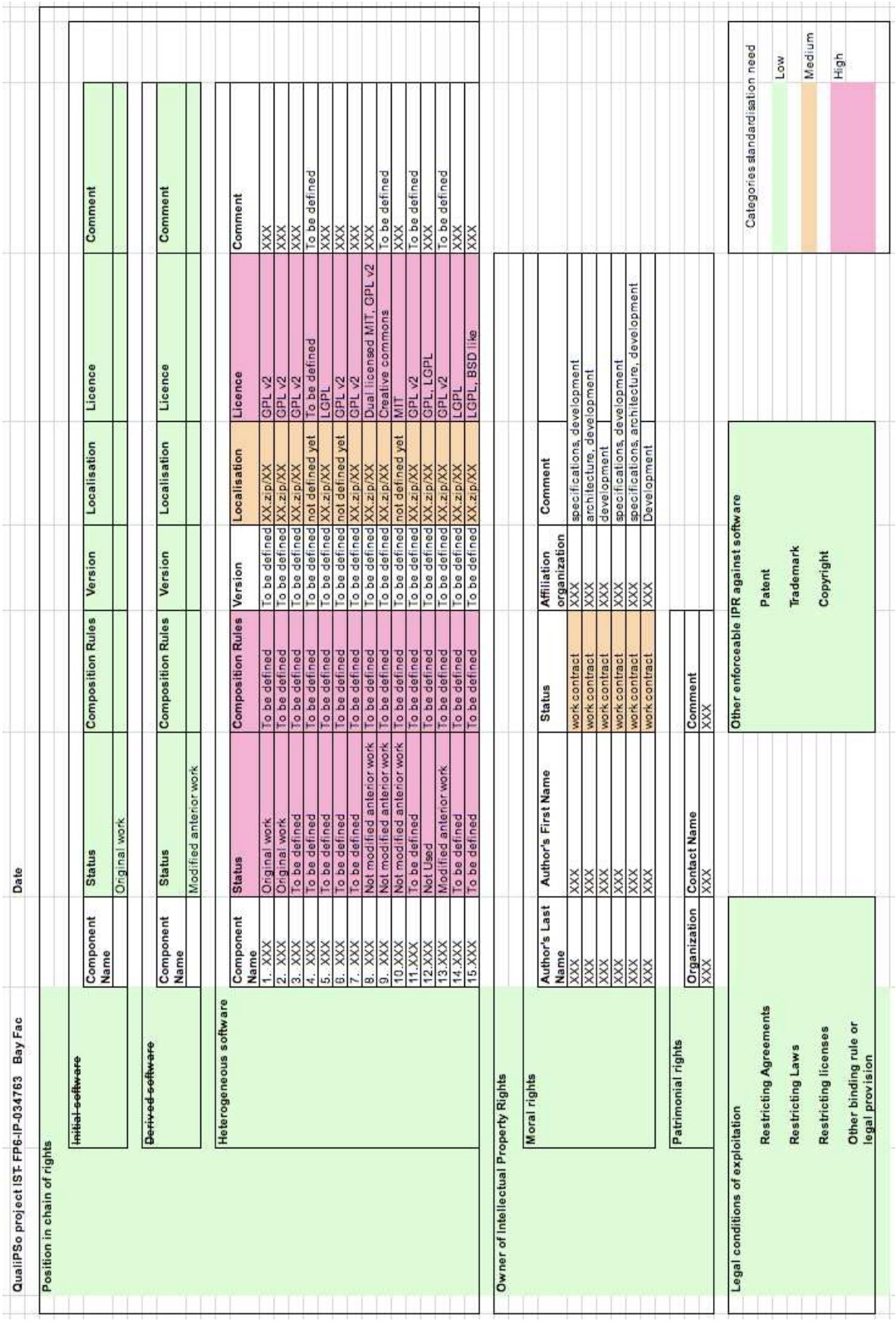

Figure 8 - BayFac Software Legal Status representation

The compiled results of the various developments and services that were audited using the FOSSology tool are presented in an internal report: “Compiled results of PALETTE audit with FOSSology tool_v1”, available on the BSCW server. All references to specific free and open source licenses that were identified by the auditing tool are included in this document, which will continue to evolve until the end of the project.

5 Recommendations

The process used for BayFac Software is extended to other PALETTE softwares. This work is in progress. However, although this preliminary work is not completed yet, we may draw observations at this stage.

First, this automated audit enables the identification of components redundancy, as well as unused component presence in the release package, which leads to corrective actions and improvement of the release package quality. This was the case for BayFac software service. We can therefore say that the quality of the development benefited from the analysis.

Second, this automated audit allowed discovering some possible issues in term of licensing interoperability. The global architectural vision of the license for the considered service needs to be compatible with the licenses of each of the individual elements of code considered.
This automated assessment would benefit from being done regularly from then on.

The choice of a license is bound by two separate obligations, as stated in DMAN08:
It allows

- “to determine if the exploitation schemas expected for the PALETTE services are compatible between them (legal analysis to be done…)”
- “to identify if the “legal statement” of the components allows these exploitation schemas. For this, we need a description of this legal statement on one hand (on the basis of answer to the questionnaires), and an analysis of the contents via dedicated tools for source code analysis. If problems are identified, we will be able to consider corrective measures, or to adapt the exploitation schema.”

This implies that an automated audit alone is clearly not enough. Out of the two auditing processes from QualiPSo used, clearly the questionnaire was the most useful, as it focused on legal and strategic elements that cannot be apprehended through the automated FOSSology treatment.

6 Conclusion

The collaboration of the PALETTE consortium with the QualiPSo project was highly beneficial. It helped create a tangible strategy in term of juridical and exploitation strategy.

The assessment processes also helped raising awareness on issues related to free and open source licenses which are far too often weakly considered. Open source licenses are not compatible with each other because of a sole consideration of them being “open source”. The open source licensing scheme describes a very specific framework, but which can be implemented in licenses in many ways, and unfortunately sometimes in incompatible ways.

The effort that was invested in studying the juridical security surrounding the PALETTE services code is an important step in terms of development management maturity. PALETTE services are based on clear and sustainable pre-existing code, each managed under licenses which are compatible with the global architectural licensing choice. This implies that the exploitation plans which shall be done for each service and exposed in a following report will be sustainable from a juridical and legal standpoint.

This sustainability is also a really important asset for community management. But as exposed previously, free and open source developments are a continuous and dynamic development effort. New versions of software might be licensed under different terms. This implies that a continuous effort in terms of juridical security to follow the licenses of used components will be required.

The various Open Source strategy reports, by raising important issues and a global awareness to licensing difficulties among all of the PALETTE actors is therefore to be considered as a successful investment within this project. The PALETTE team and the authors of this report would like to express their deepest thanks to the members of the QualiPSo project for this result.

Annex I - Specific open source licence model adopted for each of the products of the project together

The following table specifies, for each PALETTE product, the current version number and the respective license. It is followed by comments addressing risk issues.

PALETTE services		
Service	**Version**	**License**
Amaya	11.1	W3C
LimSee3	2.1	CeCILL
DocReuse	3.0	GPL v2
CoPe_it!	3.1	GPL v2
eLogbook	3.0	MIT
Bayfac	3.2.2	GPL v2
SweetWiki	2.0	CeCILL-C
Portal	2.0.1	GPL v2
Support services		
CroSSE	1.0	GPL v2
CAKB	1.0	GPL v2
OpenId Server	1.0	GPL v2
ServiceBrowser	1.0	GPL v2
Keces	1.0	GPL v2
PRep	1.2	GPL v2
PSR (Palette Service Registry)	1.1	GPL v2
Other components		
Generis (*)	4	GPL v2
Corese	2.4	CeCILL-C
SeWeSe	1.5	CeCILL-C

(*) Used by BayFac

Amaya

Regarding potential of further developments by research institutes, it could be worth noting that Amaya will continue to be maintained, developed and distributed in exactly the same conditions as during the Palette project, just with a bit less manpower. INRIA is still committing about 80% of IrèneVatton's time and W3C dedicates a part-time engineer (about 40%) to the project. No significant risks related to the evolution of technologies. The adopted standards for Amaya tools are still up to date and strongly established in their domain (web languages and protocols). It is reasonable to estimate that they will last for years.

Amaya will continue and "live its own life". However, this will not compromise its ability to interoperate with other Palette services, as all interactions are based on well established standards whose role is precisely to guarantee interoperability. Moreover, developing the WebDAV protocol in Amayais under wayfor offering one more way to interact with other services, in particular Palette services.

Limsee3

For the moment, INRIA Grenobledoes not have significant resources for maintaining and developing LimSee3, but is looking for new funding opportunities. As for Amaya, no significant risks related to the evolution of technologies used, which makes possible its evolution by third parties.

DocReuse

From an EPFL perspective, the DocReuse PALETTE service evolution is depending on research funding opportunities. It depends on the success of research proposals addressed in the next future. One of them is in relation with a proposal that will be addressed to the Swiss National Foundation of Research in September 2009 addressing the subject of "Collaborative Editing of Resources over the Internet"; it takes into account the reuse of existing "quite-structured" resources and their integration in the targeted collaborative editing environment. Also, no significant risks related to the evolution of technologies used, which makes possible its evolution by third parties.

About CoPe it!

As far as potential of further developments are concerned, efforts related to further development of CoPe_it! can be classified into two categories: efforts that depend on future funding and efforts that are being carried out independently of funding. With respect to the former, CoPe_it! is planned to be used in the context of new EU funded proposals that need to support collaboration and decision making in diverse application domains. With respect to the latter, the development team of CoPe_it! continuously fixes bugs that are being identified and makes minor improvements in order to ensure its functional presence on the Web.

With respect to evolution of technologies, CoPe_it! has been developed based on well established Web standards (e.g. XML, OpenID); hence, the tool is relatively immune with respect to technological evolution.

Regarding risks linked with the fact that some services may live their own life, we argue that the biggest risk lies in invalidating the interoperability features with the other PALETTE tools. However, in CoPe_it! the design decision to base all interoperability efforts on well established standards makes the problems surfacing from such isolation manageable. Although development efforts are required to re-establish interoperability, these efforts will be small.

eLogbook

eLogbook is currently distributed as a freeware and should later be distributed with a MIT license. These license schemes should enable both EPFL, EPFL members, as well as third parties to develop business models integrating eLogbook. The most promising models are related to hosting communities (hosting fees for advanced features, free for basic services) and helping corporate companies to integrate eLogbook in their own knowledge management and learning management systems (consultancy fees).

The future of eLogbook is fully guaranteed for the next 4 years, as EPFL is part of the new Integrated project ROLE in which eLogbook will be further developed and enhanced.

eLogbook was developed from scratch in the framework of the PALETTE project with the latest technologies available. So, its current implementation will survive without problem for the next five years. Later, new implementation can be considered, while keeping its core innovative features, i.e. its contextual 3A model and view, as well as the associated multi-nodal recommendations algorithms.

Software “Darwinism” is at the core of the PALETTE philosophy. So, the users will decide if eLogbook has or not an added value for them. However, thank to the participatory design approach we will still apply in the ROLE project, the future version of eLogbook should keep in line with future user expectations.

Bayfac

The exploitation and licensing strategy for Bayfac was decided to be Open Source. In order to do so, and to tackle any potential legal risk, the CRP Henri Tudor has made sure only to select compatible-licensed components. The final product, Bayfac, is licensed under the GNU GPL v2.

SweetWiki, Corese and SeWeSe services

To minimise the licensing risk with SweetWiki, Corese and Sewese, we made these services free software. They are subject to a CECILLlicense (CECILL = « CEA CNRS INRIA logiciellibre »), a license defining the principles of use and dissemination of Free Software in conformance with French law, following the principles of the GNU GPL.

Further developments of the SweetWiki, Corese and Sewese services are currently performed in the context of new projects following PALETTE.Further developments are also envisioned in the context of forthcoming projects.
The SweetWiki, Corese and SeWeSe services are developed using the most possible web standards proposed by the W3C. Among the advantages of using such standards is to guarantee the sustainability of the components of the services designed with these standards.
The SweetWiki, Corese and SeWeSe services evolve in the framework of projects where they must interoperate with other services rather than operate in isolation. So we most often face the interoperability problem than the isolation problem.
The policy followed in developing SweetWiki, Corese and SeWese is to seek complementarities between these services and the services existing or under development in other institutions.

Portal
The PALETTE portal continues its development under a new name: myWiWall (aka. "my widget wall"). It is currently available in source code form, with a software installer program, on a public Google code repository
(http://code.google.com/p/myWiWall/)
under a GPL2 code license. We have created different channels to advertise it and to support its evolution.