Technology White Paper

Managing the Risks of Open Source Software

The use of Free (as in 'Freedom') and open source software (FOSS[1]) has swept into the whole software business and is growing rapidly. The same is true in Alcatel's products and solutions. This is because the benefits of FOSS usage respond to the pressures of the telecommunications and IT markets. However, whereas mature FOSS exists, the management of its legal aspects has not reached a similar level of maturity. Alcatel has put in place an internal process for selecting mature FOSS and correctly handling the legal aspects throughout the company. Alcatel also actively contributes to the evolution of the FOSS market place.

[1] Free and Open Source Software, also FOSS, is software which is licensed to grant users the right to study, change, and improve its design through the availability of its source code. It is important to note that Free refers to 'freedom' rather than 'free of cost.'

ALCATEL

Introduction 1

Why Open source? 1

Preconceived ideas 2

Real benefits 2

Open source risk management 3

Find the license! 3

Define the nature of the software and how to use it 3

Liability 4

Process for handling open sources in Alcatel 4

Conclusion 4

Abbreviations 4

References 5

Biography 5

Introduction

The recent evolution of the telecommunications and IT markets has promoted the use of Free and Open Source Software (FOSS[1]) in the telecommunications industry.

1. Increased competition in the telecommunications industry leads to the need to reduce the cost of telecommunications solutions.
2. The necessity to cut time to market requires the use of mature, standard software.
3. Telecommunications equipment manufacturers are shifting increasingly to the provision of services – leading to the replacement of proprietary, low-level software by standard solutions.
4. The merging of the IT and telecommunications worlds necessitates the adaptation of telecommunications solutions to the short lifecycle of software.
5. The instability of software markets pushes users to select long-term solutions.
6. Finally, the momentum surrounding FOSS has fueled the birth of a strong FOSS ecosystem that, a few years ago, was limited to the Linux operating system, but now covers technologies such as database or middleware.

The growing use of FOSS allows the industry to cope with these market developments.

On the other hand, the use of FOSS necessitates the management of risks that are not present with proprietary software mainly related to legal aspects. The origin of FOSS is generally not guaranteed by a company, and this exposes its distributors to the possibility of third-party intellectual property rights (IPR) infringement. FOSS licenses are sometimes very complex, and each might be an amalgam of a variety of licenses (when one FOSS module embeds one or several other FOSS components, for instance). For example, a Linux distribution comes with hundreds of different FOSS components, each with a different license. Most of the FOSS comes with licenses that disclaim any liability, but this disclaimer is not valid under certain laws, and the distributor can be liable for a problem linked to the FOSS itself or for IPR infringement.

So the use of FOSS in products necessitates very careful handling. The objective is to avoid FOSS whose license conditions are too constraining, as well as to mitigate risks. This paper presents the benefits of FOSS usage for Alcatel and its customers, the various aspects that must be considered before their use, and the Alcatel process set in place to mitigate the risks.

Why Open source?

A few years ago, very little FOSS was mature: Linux, GNU software, etc. The FOSS business model was not very well understood by industry. And fears of risks, security breaches, lack of industrial credibility, etc. made industry reluctant to use FOSS.

Today, major companies such as IBM, HP, Intel, Sun, Alcatel, Nokia, Ericsson, and more have invested in open source. The viable business models for open source are now well known, risks are known, and mitigating solutions exist. As a consequence, many mature software technologies such as operating systems, database management systems, middleware software such as Corba, Java or Web technologies, protocol stacks, etc. take advantage of the FOSS model with large open source communities. Each single product has dedicated vendors, users, developers, testers, integrators – a full eco-system. Some FOSS have become a de-facto standard, such as Linux, or a reference implementation of a standard, such as TAO or JacORB for CORBA, JBoss for J2EE, Apache and Tomcat for Web servers, SNMP4J for SNMP. Many lead their specific markets after a

The hitchhiker's guide to Open Source

What is FOSS?	Richard Stallman who created the Free Software Foundation (FSF) in the eighties stated the purpose of FOSS as being able to control and adapt the software that we use. Software components should allow reverse engineering and allow the user to modify it. For that goal, all software must be provided to its users in source form.
What FOSS is not?	It is not necessary coming for free. Some FOSS comes with some type of fee. All software that comes under source form have not necessary an open source license (e.g., some licenses do no allow to modify the source code).
GPL License	The most popular FOSS license format, the so-called GNU General Public License (GPL) had been created to protect FOSS in its copyrights and usage. All modification done to source code in GPL must be made available in GPL to the code users. If some proprietary code is a derivative work from GPL software, the proprietary code becomes GPL. We call this a contaminating effect which we must carefully control to avoid that Alcatel's intellectual property cross-influences that of FOSS.
Other Licenses	A lot of other FOSS licenses evolved over time. Some license models are not compatible with each other (licenses represent people philosophy on software business model). These differences in license models and sometimes strange obligations demand strong governance before using any specific FOSS component in our products.
Open source community	Most FOSS is generated by one person or a group that makes the software available on Internet. External contributors provide bug corrections or additional features by proposing source code to the original team. The team accepts or rejects contribution and it controls the official release of the software. They become what is called "benevolent dictators".
FOSS distributors	A new supply chain has evolved over the past years with independent companies which distribute most of the mainstream FOSS. They work with the open source community and support and maintain the FOSS. They provide stability to release by synchronizing regularly (but not too often) with the open source official release. This specific channel model of FOSS creates effort and thus adds to the cost of usage, but assures the same or typically better protection to the end-users of FOSS within a product or solution that a single supplier would provide.

short time due to the obvious benefits of such rich and dynamic eco-system. However there are some preconceived ideas concerning what are the advantages of FOSS that we want to remove from this paper.

Preconceived ideas

Preconceived idea 1:

"FOSS is cheap software!" When it comes to industry strength software with sustainable maintenance, dedicated suppliers are needed. The total cost of FOSS procurement and usage (license, support) is not much lower than that of proprietary software. The advantages of FOSS usage lie elsewhere: reduced dependence on supplier terms and conditions, better competition among distributors, a long-term software solution even if a supplier changes, broad user basis for mainstream FOSS, which assures fast turn-around time for corrections, new features, security fixes, etc.

Preconceived idea 2:

"FOSS can be used freely." Not true. FOSS follows specific license models, which vary significantly depending on product, user, and usage. FOSS license management is more difficult than that of proprietary software because of its many owners. Some license conditions make sustainable commercial usage in Alcatel products impossible. Alcatel sometimes negotiates specific schemes with license owners to optimize the benefits to customers.

Preconceived idea 3:

"FOSS can be downloaded and used as is." Just downloading FOSS carries many risks related to configuration, stability, or security. It has been made very clear to Alcatel engineers that they must handle FOSS just like any other proprietary software component. They apply the same care and only include FOSS that has been approved and qualified and is good enough to sustain our high quality standards.

Real benefits

The real benefits of FOSS usage in order of priority are the following:

Durability.

The software market is fast-moving and many supplier companies face uncertainties. Alcatel is careful to select FOSS with large open source communities. They are lasting solutions that are not subject to abrupt end-of-product-life. They are also supplier-independent, which avoids the need to migrate from one proprietary solution to another, and that helps to reduce product costs.

Quality and responsiveness to defects.

The source code of mainstream FOSS is scrutinized by a large community of software specialists. These tend to provide rapid fixes to defects and assure overall high quality and stable releases of the FOSS. The manpower dedicated to mature FOSS is greater than that allocated to proprietary software. Fixes to correct problems can be applied to FOSS and are not subject to the inclination or resources of a single supplier company.

Fair suppliers.

Distributors of a common open source (e.g. Linux or TAO) differentiate from each other by their business model, their terms and conditions and their service level, not by the product itself. This increases competition. Alcatel products are based on **FOSS supported by distributors.** If the distributor terms and conditions become inappropriate, it is easy to change distributor. This ability to pick and choose distributors helps to build a community of fair suppliers.

ISV integration.

The different pieces of software in an Alcatel product need to be integrated with each other. Independent Software Vendors (ISV) assure this integration by working with each other. Sometime the integration comes with a cost for Alcatel. Using popular FOSS avoids these costs. For instance, most ISVs assure the integration of their software with Linux from Redhat or SuSE, or with MySQL database.

Managed cost for long-term support.

With the merging of the telecommunications and IT worlds, the life-cycle of telecommunications products must be aligned with that of software. Typically, telecommunications product releases have a life-time of five or more years, while software releases have a time scale of around 18 months. Using FOSS in telecommunications products avoids costly contracts for long-term support by in-sourcing the long-term support; this is possible because the source code is available.

Faster to market.

Using wide-spread, standard software components, well known by engineers and integrated with each other, reduces the amount of integration among telecommunications products needed to form end-to-end solutions. It speeds up the learning curve and minimizes the various problems outlined above, allows the rapid development of new products, and accelerates market introduction.

Shaping the Open Source Software Market - the CORBA example

The Alcatel preferred CORBA supplier (ITU-T and Telemanagement Forum standard from the Object Management Group (OMG) for Network Management) was acquired in 2001 by one of it's competitors, resulting in dramatically changed licensing conditions for Alcatel. As a result, the company initiated a survey of CORBA Open Source Software and found that many lacked sufficient mass to bring an industrial-strength product to market. Finally, contact was made with a company developing services on top of proprietary CORBA systems; they were persuaded to support/distribute two Open Source Software CORBA products: JacORB (Java, general-purpose-based) and TAO (C++, real-time and general-purpose). A business model was co-developed with them to include specific enhancements that were part of Alcatel's funding proposal. This resulted in Alcatel signing a 3-year contract (meanwhile extended), during which all Alcatel Corba-based solutions were migrated. Contacts with other interested parties (Alcatel customers, other NEPs) were organized, strengthening the further industrialization of the Total CORBA Solution (TCS).

Tighter security.

A MITRE and US Department of Defense study concluded that FOSS improves security. The review of source code by many people compensating for the fear that security can be breached because source code is available.

Popular acclaim.

Last but not least, FOSS is popular. Software engineers and students alike have embraced the FOSS philosophy. Popular FOSS is known by students, shortening the learning curve, and engineers often use it at home. This has a positive impact on the working atmosphere and productivity.

In short, using mature FOSS dramatically reduces costs (even though it is not free), while improving quality and cutting development time.

Open source risk management

Alcatel decided to introduce mature FOSS bottom-up. This means that FOSS is used for platforms and platform elements, such as operating systems, stacks, middleware, etc. At the same time, Alcatel has worked to accelerate the maturing of FOSS, for instance, through participation in the OSDL Carrier Grade Linux initiative.

Alcatel & OSDL Carrier Grade Linux

Alcatel is one of the nine founders of the Carrier Grade Linux working group of the Open Source Development Labs (OSDL), together with Intel, IBM, HP, Cisco, Ericsson, MontaVista, RedHat and SuSE. It was created in January 2002. The goal was to strengthen Linux for its use in the telecom industry and to adapt it to embedded system developments. As a result, Linux has today extended its market share to embedded system in which soft-real-time behavior is necessary; it has been widely embraced by the telecommunications industry.

On the other hand, using FOSS implies additional risk management. The Alcatel policy is to use FOSS distributors that provide support, packaging, some legal liability, etc. However, distributors do not always exist, and when they exist they do not provide the same guarantees as companies that fully own the software. Before selecting a FOSS module, three majors legal aspects need to be controlled: determining the origin of the FOSS, ensuring that the usage of the FOSS is compatible with its license, and mitigating the liability risks.

Find the license!

Determining the origin of FOSS and the copyright and license to apply is not an easy task. Difficulties are encountered in 30% to 40% of cases. Finding the right license or composite licenses linked to a FOSS is crucial because it defines the rights (and obligations) to use, copy, modify and distribute the FOSS.

Software with no proper license does not mean that the user is free to interpret how it can be used, copied, modified or distributed. Instead, the original software owner could still establish a license in the future - with all kinds of implications for terms and conditions. A common mistake is to consider software without a license as "public domain" software (i.e., software with no restriction of use). Public domain software is software for which the copyright holder (and not somebody else) has explicitly declared the software as "public domain" in its copyright. Today, some Linux distributors still distribute some software that has no clear license. Alcatel requires its Linux distributors to provide the list of software in its products, their licenses, and the nature of the software (standalone code, library, applet, etc.).

One FOSS module can use several FOSS components internally, each with different licenses. And this can be recursive. The licenses must be compatible with each other, according to the link between software components. For instance, a Linux distribution can comprise from 300 to one thousand software components of various origins with different licenses and obligations.

Because some people who redistribute FOSS are not always rigorous, the license announced on a Web page may not be the exact, real license that comes with the software. It is often necessary to check the source code to find the exact license.

There are many other such traps, for example, licenses that change with a new FOSS release. For instance, Mibble is GPL in version 1.0, 1.1, 2.4 and 2.5 and LGPL-like from 1.2 to 2.3. Or FOSS may have license dependencies with other FOSS licenses because they are a translation from one programming language to another (for instance, Jtidy, which is a Java port of HTML Tidy). Experience shows that it can take a great deal of effort to find the exact license of a specific FOSS and optimize its usage. This effort, however, is well invested, because it can assure sustainable benefits from the FOSS component.

Define the nature of the software and how to use it

FOSS licenses can have a contaminating effect on a company's own software development. For instance, any derivative work based on GPL FOSS becomes GPL itself! Among other things, this means that derivative work must be distributed in source form, and that can lead to the loss of competitive advantage. The definition of "derivative work" is "software running in the same address space", i.e., the software necessary for debugging the whole system. An example is the Linux kernel, which is GPL. Developing an application running on a separate address space from the kernel avoids contamination, while plugging a proprietary driver into the kernel implies driver GPL contamination.

Use of FOSS can also be subject to trade regulations. For instance, some encryption software necessitates export licenses from local authorities for each country of use, while "mass market" encryption software (OpenSSL, Java Virtual Machine JCE basic encryption package) may not. When there is a supplier for the FOSS, these procedures can be partly handled by the supplier. But for most FOSS without dedicated distributors, the issue needs to be handled by the re-distributor or the user of the FOSS.

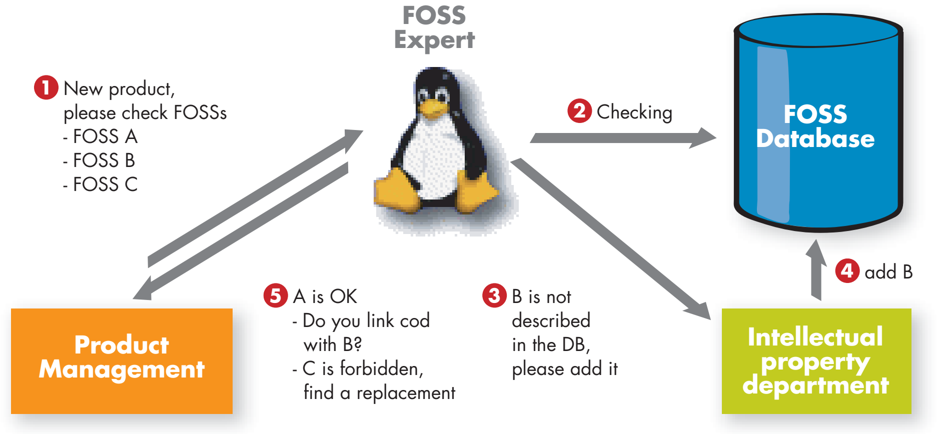
In consequence, and depending on its license, the nature of FOSS (standalone software, library, Java applet, etc.) must be carefully examined, together with the potential relationship with proprietary code, to determine the legal and business implications.

Some licenses such as GPL or LGPL (Lesser GPL) demand that any modification to the GPL/LGPL code be made publicly available in source form when distributing it. That means, use of the FOSS must also be taken into account when selecting it. Alcatel has set up a process to communicate its FOSS contributions or changes.

Liability

Despite the fact that most FOSS disclaims any liability on the software, (re-) distributor liability cannot be waived under legislation prevailing in many countries. Liability exposure might result from defects in the FOSS. Contrary to proprietary software, liability cannot be assumed by a supplier, so the (re-)distributor must often assume a risk similar to that of distributing proprietary code. Alcatel mitigates this risk by selecting FOSS with large communities, which guarantee code quality, by selecting software that is stable, or by using a distributor that provides a certain level of guarantee.

Figure 1: FOSS management process


FOSS Expert
1 New product, please check FOSSs
- FOSS A
- FOSS B
- FOSS C
2 Checking
FOSS Database
3 B is not described in the DB, please add it
4 add B
5 A is OK
- Do you link cod with B?
- C is forbidden, find a replacement
Product Management
Intellectual property department

Copyright infringement can happen when some third-party software has been included in the FOSS. Since the exact origin of the FOSS is not clear, the risk is greater than for proprietary software and can be discovered when source code is available. Through careful selection of software and close examination of licenses, Alcatel reduces this risk. Changing the infringing code is a way of mitigating the risk.

Patent infringement by a FOSS is the third liability risk. Again there is always the possibility of changing the infringing code and mitigating this risk by selecting well-known and widely-used open sources. The risk is more limited for companies like Alcatel with a large patent portfolio, which can use patents as a protective means.

Process for handling open sources in Alcatel

Alcatel maintains a centralized FOSS-support database on its Intranet describing legal aspects of open source. All entries have been validated and qualified, as indicated above. Each entry contains the exact origin (URL) and version(s) of the open source and the exact origin and version of its license. Most open sources reference well-known licenses (GPL, LGPL, BSD, Apache, MIT, Mozilla, etc.) for which legal risks and obligation are described in detail in the database, but specific licenses are also described. Entries also reference dependencies on other FOSS components and the situation regarding encryption and patents (when known). Each entry presents a global recommendation on the FOSS according to the use planned, the level of risk, and the precautions to take.

Training is organized in Alcatel to form open source experts. Aside from general legal information on FOSS usage, experts are trained on the most current license schemes. Names of trained experts are also kept in a centralized database. Finding the “nearest expert” in the organization is therefore made easy. Before the start of Alcatel product/solution development, the FOSS planned for use and how it is to be used must obtain approval from FOSS experts. When the FOSS is not described in the database, the intellectual property department takes over. This permits decentralization and at the same time maintains the quality of the FOSS selection.

Conclusion

The use of mature FOSS has many advantages that address current software and communication market needs. The software market is unstable and software users demand long-term solutions and minimum dependence on specific suppliers. Alcatel’s use of FOSS enables cost reductions, faster time to market, and improved quality and security

FOSS brings with it a number of risks that are not yet satisfactorily handled by most companies. The Alcatel process for handling FOSS has raised awareness of legal aspects throughout the company and provides guarantees that these aspects are correctly handled. Alcatel is also lobbying FOSS suppliers, when necessary, to better handle these aspects so as to reduce risks.

Finally, Alcatel is actively working at nurturing the FOSS market and business models. It is doing this by its actions through OSDL Carrier Grade Linux or on CORBA, but also by cooperating with market stakeholders. Alcatel’s strategy is clearly to accelerate the maturing of FOSS.

Glossary of terms and abbreviations:

BSD	Berkeley Software Distribution license
CORBA	Common Object Request Broker Architecture
FOSS	Free/Open Source Software
FSF	Free Software Foundation
HTML	Hypertext Markup Language
IPR	Intellectual Property Rights
ISV	Independent Software Vendor
IT	Information Technology
ITU-T	International Telecommunication Union – Telecommunication standardization sector
J2EE	Java 2 Enterprise Edition
GNU	Gnu is Not Unix
GPL	General Public License (GNU)
LGPL	Lesser (or Library) General Public License (GNU)
MIT	Massachusetts Institute of Technology license
OMG	Object Management Group
OSDL	Open Source Development Lab
SNMP	Simple Network Management Protocol
SSL	Secure Socket Layer
TAO	The ACE ORB
URL	Uniform Resource Locator

References

[1] Use of Free and Open Source Software (FFOSS) in the U.S. Department of Defense. MITRE report MP 02 W0000101, January 2, 2003.

[2] David A. Wheeler: Why Open Source Software / Free Software (FOSS/FS, FLFOSS, or FFOSS)? Look at the Numbers! Revised May 9, 2005, http://www.dwheeler.com/FOSS_fs_why.html.

[3] Salvino A. Salvaggio: Open source: A r/evolution in the software industry. June 2004, http://www.salvaggio.net/index.php?section=3&page=Articles&mode=fulltxt&nid=98&PHPSESSID=7aa9a0242f6ee093f75fef46e3c95662

[4] Ruffin, M. and C.Ebert: Using Open Source Software in Product Development: A Primer. IEEE Software, Vol. 21, No.1, pp.82-86, Jan. 2004.

Michel Ruffin obtained a Ph.D in computer science in 1992 at the University Paris VI (France). From 1993 to 1995, he took a post-doctoral position at the University of Glasgow (Scotland) in the area of distributed object-oriented systems. He joined the Alcatel Research Center in 1996 to work on middleware for telecommunications. In 2001, he joined the Alcatel CTO team to coordinate the technical procurement of software components for Alcatel products. He has chaired the Telecom group of the OMG since 1999 and is a member of the OSDL Carrier Grade Linux Working Group steering committee. In 2000, he received the OMG's Distinguished Service Award and since 2001, he has been a distinguished member of the Alcatel Technical Academy. He has been serving as expert on research projects for the European Commission since 2002.

Christof Ebert, today director of R&D processes and tools at Alcatel, Paris, has directed engineering and IS projects and process improvement initiatives at Alcatel for more than a decade. He received his Ph.D with honors in Electrical Engineering from the University of Stuttgart, where he today lectures on software engineering. He is a senior member of IEEE Computer Society and the editor of IEEE Software's Open Source column. His research and consulting covers innovative R&D strategies and software management He is a member of the Alcatel Technical Academy.