Introducing Linux and open source

Lotte Mygind, Rune Hylsberg Jacobsen and Oskar Swirtun

Ericsson is adapting its products and work processes to take advantage of opportunities afforded by open-source software. In particular, the Linux operating system (OS) has become an important component for Ericsson.

The authors describe the use of Linux and open-source software in Ericsson products and how they benefit Ericsson's customers. They also touch on the characteristics of open-source software and present a new Linux-based building block for Ericsson's Integrated Site applications.

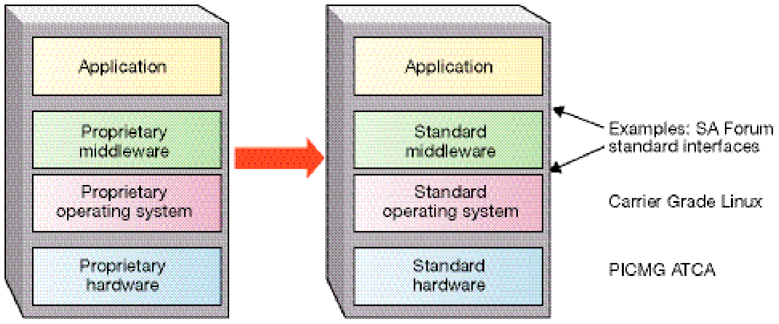
Background

The communications industry is moving toward systems that are based on modular and common, off-the-shelf hardware and software components. The underpinnings of this transition are open standards and architectures (Box A, Item I).

Open architectures and the standardization of interfaces between associated components are nothing new. What is new, however, with regard to the technological trend to strengthen the notion of openness, is the use of open-source software, which implies that many components contain open-source software supplemented with common, reusable components.

This trend is motivated by vendor and operator emphasis on reducing the costs and risks of development. Open standards ensure portability and interoperability between various components from different providers. For example, the open standards described by organizations such as the IEEE, IETF, and SA Forum are already widely accepted and in use.[1-3] Likewise, the Linux OS has taken on an important role as a carrier-grade operating system. Figure 1 illustrates the shift toward standards-based layers of composition.

Figure 1
The transition from proprietary to standardized solutions. The Service Availability Forum standardizes interfaces between middleware, the operating system, and applications. The OSDL drives the implementation of carrier-grade requirements in Linux; hardware standardization is carried out in the PICMG ATCA.



What is open source?

The term *open source* refers to software whose underlying source code is either delivered together with the software or can easily be obtained, for instance, via the internet.[4] Anybody may use, modify, redistribute or sell open-source software. The idea behind an open source is that the process of having many people read, modify, test and debug it should also improve the source. The mere availability of source code is not enough to qualify as open source, however. For example, Microsoft's shared source prohibits commercial use of modified code. The Open Source Initiative (OSI) maintains a formal definition of open source (Box A, Item II).

Proprietary licenses, by contrast, often put restrictions on the use and distribution of software. They generally also deny users access to the source code, and users are frequently required to register and pay for the software.

Some open-source licenses require that modified versions of the software must be licensed in a special way to prevent it from being used as proprietary, closed-source software. A prime example is the GNU General Public License (GPL), which stipulates that works derived from it must also be licensed as open source under the GPL license (Box A, Item III). Because the GPL transfers to derived works, software development companies have complained that the use of GPL software could transform their products into freeware that can be copied by competitors.

The definition of free software is slightly more restrictive than the definition of open source, but the differences are minor. The main difference lies in the political and philosophical approach of the Free Software Foundation (FSF) versus the pragmatic approach of the OSI, which promotes open source for its technical qualities. "Free" refers to freedom of use, not cost. The FSF maintains the definition of free software.

In recent years, there has been greater understanding of GPL in the commercial world, and the legal interpretation has grown clearer. As a consequence, GPL software is now found in numerous commercial products.

The processes used in open source development projects differ from those used in traditional commercial software organizations. In fact, some people claim that these processes are superior to traditional development processes (Box A, Item IV).[5]

Operator benefits

Open-source software offers the following immediate advantages:

- Availability of source code: The source code is readily available and can be modified to meet specific requirements. For Ericsson and its customers, this means shorter time to market. What is more, Ericsson or the open-source community can fix bugs more quickly, making maintenance more effective.
- Vendor independence: Because a lot of open-source software is supported by a variety of companies, Ericsson and its customers need not be locked in to a specific supplier.
- When Ericsson uses software from open-source projects, the associated processes contribute toward high quality and facilitate rapid development of new features (Box A, Item V).

Legal issues

The use of open source software in Ericsson products imposes similar legal issues as proprietary software. Internally, within the development organizations, the use of GPL software as well as software under similar licenses poses an additional risk because complying with licenses can be difficult and might ultimately force the organization to publish the source code of software that was not intended to be open source. Also, as is the case with any software, there is the issue of intellectual property (patent or copyright) infringements. Ericsson evaluates open-source software before incorporating it in commercial products.

Many commercial vendors offer to protect their customers against claims of intellectual property infringement. Notwithstanding, the question of potential infringements in open-source software is as yet an unresolved issue. It should be noted, however, that the same can be said of proprietary software: despite comprehensive efforts to check for patents, commercial vendors may not be able to guarantee that a patent does not apply. Consequently many vendors refuse to take full legal responsibility for their proprietary software.

Linux

A Linux distribution (also called GNU/Linux distribution) is an operating system

BOX A

I. HISTORY OF OPEN SOURCE

- The early days: Collaborative software development is not formalized.
- 1983: Former MIT employee Richard Stallman starts the GNU project. The project has since created an enormous body of open-source software.
- 1985: Stallman forms the Free Software Foundation (FSF), a sociopolitical organization that promotes what it calls “free” software.
- 1991: Finnish computer science student Linus Torvalds invites people to contribute to a free operating system and makes the first release of the Linux kernel software.
- 1998: The term *open source* is coined.
- Today: Many vendors provide support for open-source software, such as Linux distributions, which typically bundle the Linux kernel with GNU programs and a graphical interface.

II. THE OPEN SOURCE INITIATIVE (OSI)

- OSI is a non-profit organization that works to promote open-source software.
- OSI maintains a definition of open source.
- OSI maintains a list of OSI-approved software licenses (at present, more than 50).
- OSI states that “open source promotes software reliability and quality by supporting independent peer review and rapid evolution of source code.”

III. GNU GENERAL PUBLIC LICENSE (GPL)

- The Free Software Foundation published the first version of the GPL in 1989.
- GPL is the most common open-source license in use.
- The Linux OS kernel is licensed under GPL.

IV. WORK PROCESSES IN LARGE, OPEN-SOURCE PROJECTS

- Frequent releases make changes available to users at an early stage.
- Parallel development yields different solutions, of which the best is chosen.
- Rapid implementation of new features thanks to an experimental environment and many developers.
- Peer reviews – everybody can read and find errors in the source code and contributions.
- User testing – everybody can report errors; users of the software thus provide extensive testing.

V. QUANTITATIVE PROOF

Using quantitative measurements, David A. Wheeler compared Linux and open-source software products.[6] From his study, Wheeler drew the following conclusions:

- Open-source or free software has a significant share of many markets. Examples are the Apache web server, used by more than 60% of all web servers, and the Linux OS, which is used on a large share of web servers and other servers on the public internet.
- Open-source or free software is often the most reliable software. Studies show that the Linux kernel has significantly fewer bugs than its industry alternatives. Furthermore, experiments have demonstrated that Linux systems hardly ever require rebooting even after they have run for a long time.
- In many cases, open-source or free software has the best performance. Several tests show that the performance of open-source systems matches or exceeds the performance of proprietary equivalents. In 2002, *PC Magazine* demonstrated that Linux with Samba performed excellently compared with a proprietary alternative solution.[7]
- Open-source or free software scales well both in terms of situation and project size. GNU/Linux systems and NetBSD support a broader range of hardware platforms than any other operating system. In November 2004, the Blue Gene/L supercomputer, in which Linux is the main operating system, became the most powerful supercomputer in the world.[8]
- Open-source or free software is often more secure, thanks no doubt to worldwide review. Studies show that attacks against GNU/Linux systems are less successful than attacks against proprietary equivalents, and that the majority of serious security problems do not apply to open-source software.
- The total cost of ownership of open-source or free software is often far less than that of proprietary software, especially as the number of associated platforms increases.

VI. ERICSSON CONTRIBUTIONS TO OPEN-SOURCE

- 1998: Ericsson releases Erlang under an open-source license. Erlang is a programming language and runtime environment for building distributed, real-time, high-availability systems.
- 2003: Ericsson releases the transparent inter-process communication (TIPC) protocol to open source. Ericsson designed the protocol for intra-cluster communication, and has used it in its products for several years.

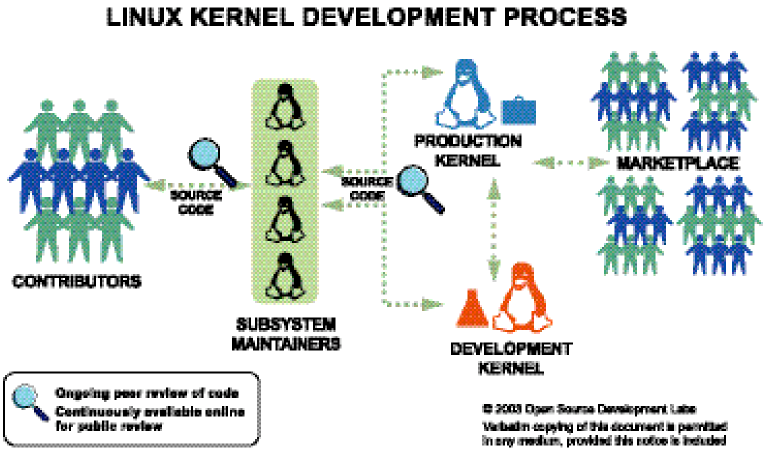

LINUX KERNEL DEVELOPMENT PROCESS
CONTRIBUTORS
SOURCE CODE
SUBSYSTEM MAINTAINERS
SOURCE CODE
PRODUCTION KERNEL
MARKETPLACE
DEVELOPMENT KERNEL
Ongoing peer review of code
Continuously available online for public review
© 2003 Open Source Development Labs
Verbatim copying of this document is permitted in any medium, provided this notice is included

Figure 2
The Linux kernel development process. New features and bug fixes are tested in the development kernel (maintained by Linus Torvalds, OSDL). The production kernel (maintained by Andrew Morton, OSDL) is the official Linux kernel.

that includes the Linux kernel developed by Linus Torvalds, GNU software, and other software (typically open source).

The Linux kernel has a large, flourishing development community and a large user base (Figure 2). The Linux kernel community implements a very specialized division of labor: specific "maintainers" have been designated for different areas, and a formal procedure screens the acceptance of contributions to official kernel versions.[9] Although the responsibility for various areas centers on a limited number of maintainers, most contributors submit only a few changes.[10] Contributor activity levels are generally very high, however, as evidenced through many postings on mailing lists and discussion forums. Contributors also submit a great deal of code.

Several projects implement test suites for Linux. The foremost of these is the Linux Test Project (LTP)[11], which is funded by a variety of organizations including the Open Source Development Labs (OSDL). Founded in 2000 by a number of commercial organizations (including Ericsson), the OSDL promotes the commercial use of Linux. It provides Linux testing, equipment and assistance to the open-source community. The OSDL also employs Linux Torvalds and the Linux production kernel maintainer, Andrew Morton.

Linux also enjoys extensive hardware support. Today, for example, it is not uncommon for hardware suppliers to release Linux drivers with their products, which is to say that Linux runs on virtually every processor on the market.

Carrier Grade Linux

In January 2002, the OSDL established the Carrier Grade Linux (CGL) working group to accelerate development of Linux for the data communications and telecommunications industries. Members of the CGL working group represent telecommunications and data communications service providers, Linux distributors, and network equipment providers.

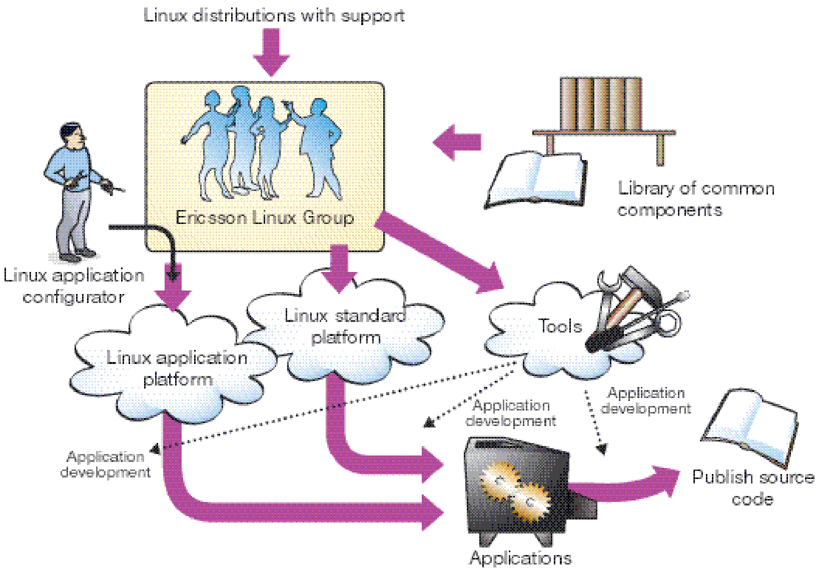
The CGL working group has successfully driven the standardization of Linux for telecommunications and data communications requirements. In October 2003, it released CGL version 2.0, and in early 2005 it certified the first Linux distribution. To date, the OSDL has registered six Linux distributions as being compliant with the CGL 2.0 specification.[12] CGL version 3.0 was published in February 2005, and Linux distributions that implement it should become available this year.

Open source at Ericsson

Ericsson contributions to open source

Ericsson has formed a steering function, the Open Source Steering Committee (OSSC),

Figure 3
The Ericsson Linux Group.



with the aim of balancing efforts to protect its intellectual property and to contribute to open source development. Before any open-source software developed by Ericsson can be released, it must be evaluated and approved by the OSSC. Ericsson's most important contributions to open-source software to date are the Erlang programming language and run-time environment and the transparent intercommunication protocol (TIPC, Box A, Item VI).

Responses to Ericsson's release of TIPC have been very positive: an open-source community has been formed around the protocol, and through the work of this community, Ericsson's representatives cooperate with numerous organizations and individuals including representatives from competing vendors. Today, Ericsson spends less time on maintaining TIPC than it did before the software was released as open source. In addition, the quality of the implementation has improved. TIPC is today an industry-standard protocol accepted and merged into the official Linux kernel.

Ericsson's open-source strategy

Ericsson's strategy is to negotiate support contracts with commercial vendors of open-source software. Components slated for use in an Ericsson product but which are lacking in adequate support from a commercial vendor must be thoroughly evaluated and approved by management. Each evaluation takes into consideration licensing terms, IPR issues, commercial use of the component, known errors, alternative components, and development activity in the open-source community.

In recent years, several of Ericsson's products have incorporated open-source software and open standards. Examples include Telecom Server Platform (TSP), Integrated Site (IS) concept, APZ, Connectivity Packet Platform (CPP), Enterprise Media Gateway, and MX-ONE Telephony System.[13-15]

Centralized Linux handling

Ericsson has formed an organization called the Linux Group to deliver and support Linux distributions for a variety of its products. An inherent danger of working with a distribution of open-source software is that one can all too easily end up maintaining a "proprietary" version of one or more distributions. By concentrating the work into a single organization, Ericsson hopes to reduce this risk. Even so, there is still the danger of creating vast overhead if development moves away from the commercially supported distribution. For this reason, Ericsson's Linux Group makes as few changes as possible to the distribution and works actively to submit its software changes to the vendor, so that the vendor can support and maintain them.

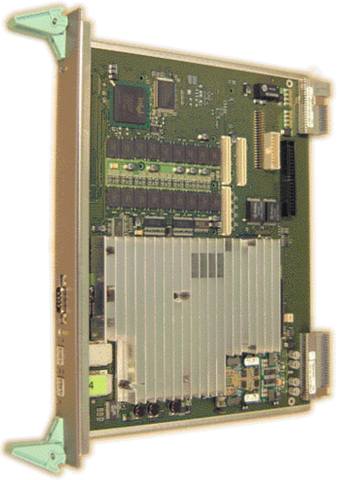
Furthermore, the group solely implements modifications in Linux that are relevant to more than one Ericsson product. Special features of a single product must be developed and maintained under the immediate budget of that product. The cost of maintaining specific changes is thus directly connected to the budget of the end product. As a result, Ericsson can clearly determine whether it wants to continue maintaining the changes or look for other solutions. Apart from reducing the number of proprietary variants of the Linux distribution, this approach

- concentrates licensing issues, and the handling of support contracts in the Linux Group;
- has led to the establishment of a central Linux software repository and update servers;
- calls for a central adaptation to Ericsson's product-handing processes;
- makes the Linux Group a support interface to Linux vendors;
- has led to the establishment of common security-monitoring mechanisms;
- has led to an increase in tools for subsequent modification of the distribution. For example, the Linux Group has developed tools for building a modified distribution and for handling the source code delivered with the product;
- has made it possible to add components (that are not otherwise part of the distributions) to the portfolio with or without support from an external vendor. The software base can thus be extended to cover additional open-source components; and
- makes it possible to identify common requirements of various products and to reuse development work.

Figure 3 illustrates the role of the Linux Group. The starting point is a handful of Linux distributions and vendor support. The Linux Group handles the distributions in Ericsson's product-handling framework. It develops a set of tools that can be used for handling the distribution and for application development. It also develops and maintains a set of common components that can be used as a supplement to the distribution.

The group has already delivered several Linux application platforms. An application platform is an operating system that can be

Figure 4
The CPBS processor board, which fits into the Enhanced Generic Equipment Magazine (E-GEM).



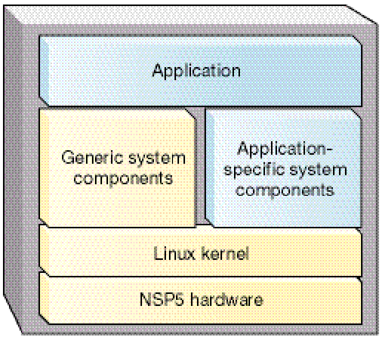

Application
Generic system components
Application-specific system components
Linux kernel
NSP5 hardware

Figure 5
The application development project (blue) and the Linux Group (yellow) are responsible for different parts of the complete solution.

used by more than one end product. In some instances, an application platform includes hardware. If this is the case, the Linux Group handles the integration of hardware and operating system. An application platform might also be a set of tools that are used to build (adapt) the operating system to the application. The Common Processor Board Subsystem (CPBS) is one example of a delivered application platform.

Ericsson's Linux Group also delivers a standard Linux platform – a distribution from a commercial vendor that has been stored in Ericsson's product-handling system along with appropriate documentation and guidelines.

Benefits of open source to Ericsson and its customers

The two main customer benefits of open-source software and open standards in Ericsson products are high quality and shorter time to market. One additional benefit of open standards is a greater range of flexibility in the finished solution – a system solution, for example, can be built from the integration of standard components selected from other Ericsson systems that

- implement the same standards;
- are sourced from a third party vendor; or
- are obtained from an open-source project.

By incorporating open-source components into its products, Ericsson can shift its focus away from traditional software development – that is, away from writing and testing source-code components. The result is a shorter development cycle and greater focus on other tasks, such as requirements analysis, supplier agreements, integration, and system verification.

Application example: Linux building block for IS

As is being done elsewhere in the industry, Ericsson uses standards-based building blocks for hardware, operating systems, and middleware. The Integrated Site, for example, is a framework for hosting applications using open-standard interfaces between network elements.[14] The primary drivers for using open-source software and open standards in IS are greater service flexibility, in terms of offering a layered framework for applications, shorter time to market, and better quality.

CPBS

The Common Processor Board Subsystem, an application platform developed by Ericsson's Linux Group, is an IS building block (or subsystem) used by the Session Border Gateway (SBG) and the EDA Multiservice Access Node.[16] The general idea is that the CPBS can reuse development in different IS application blades.

CPBS contains a common hardware platform, composed of a processor blade used in IS and TSP (Figure 4). The Linux Group is responsible for hardware drivers, boot loader and hardware integration. It is also responsible for hardware, operating system (Linux kernel) and generic system components, such as GNU software. The application projects are responsible for the application and application-specific components on top of the operating system (Figure 5).

CPBS software is a commercial Linux distribution. Additional components have been added to the software base. An accompanying set of tools and a build environment can be used to modify the operating system. For example, an application project might modify the package set of the distribution to suit specific requirements. The build tools output a set of load modules that fulfill the principles used in IS software management. The generic system components (Figure 5) include

- packages from the commercial Linux distribution – these are needed by the applications; and
- a set of common components that enable the blade to function in the IS framework. Examples of common components are a GVRP daemon for managing IS VLANs,

TRADEMARKS

- The Linux trademark is owned by Linus Torvalds.
- OSDL is a registered trademark of Open Source Development Labs, Inc.
- UNIX is a registered trademark of the Open Group.
- Microsoft is a registered trademark of Microsoft Corporation.

a mechanism for fast-link-layer failover in IS, and scripts that handle the first boot phases in the IS framework.

Current IS applications (based on the CPBS application platform) use a package selection that differs significantly from the distribution. Therefore, based on project experience to date, it makes sense to maintain a flexible solution for the CPBS so that each implementation of the application platform can be derived from a configuration of a common base of hardware and software.

Conclusion

Linux is an open-source, carrier-grade operating system built on open standards. Numerous studies show that Linux and other open-source products perform well, have good quality, scale well, and have few serious security issues. Therefore, Linux and open-source software give the telecommunications industry a way of making products and systems more cost-effective without having to compromise on quality.

Ericsson views Linux and open-source software as strategic investments. For example, Ericsson uses the Linux OS in Integrated Site (IS), Telecom Server Platform (TSP), and APZ.

To rationalize development, Ericsson implements Linux building blocks that have been designed to ease the task of implementing Linux-based applications. One such building block is the Common Processor Board Subsystem (CPBS), which is a combined hardware and software solution for developing IS application blades.

Ericsson's customers also benefit from the company's use of open-source software, in particular, because Ericsson can

- implement high-quality, carrier-grade products in less time (shorter time to market); and
- offer solutions with a greater range of service flexibility optimized for different operators' total service offering and adjusted to ever-changing business needs.

BOX B, TERMS AND ABBREVIATIONS

ATCA	Advanced telecom computing architecture
CGL	Carrier Grade Linux
CPBS	Common Processor Board Subsystem
CPP	Connectivity Packet Platform
FSF	Free Software Foundation
GARP	Generic attribute registration protocoll
GCC	GNU compiler collection
GNU	GNU's not UNIX)
GPL	GNU General Public License
GVRP	GARP VLAN registration protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IS	Integrated Site
LAN	Local area network
LTP	Linux Test Project
OS	Operating system
OSDL	Open Source Development Labs
OSI	Open Source Initiative
OSSC	Open Source Steering Committee
PCI	Peripheral component interconnect
PICMG	PCI Industrial Computer Manufacturers Group
SBG	Session boarder gateway
TIPC	Transparent intercommunication protocol
TSP	Telecom Server Platform
VLAN	Virtual LAN

REFERENCES

1. Institute of Electrical and Electronics Engineers (IEEE), www.ieee.org/portal/site
2. Internet Engineering Task Force (IETF), www.ietf.org/], OSDL [Open Source Development Labs (OSDL), www.osdl.org
3. Service Availability Forum (SA Forum), www.saforum.org/home
4. The Open Source Initiative, www.opensource.org
5. Eric S. Raymond: The Cathedral & the Bazaar (O'Reilly; hardcover ISBN 1565927249, October 1999; paperback ISBN 0596001088, January 2001)
6. David A. Wheeler: Why Open Source Software / Free Software (OSS/FS, FLOSS or FOSS)? Look at the numbers!, www.dwheeler.com/oss_fs_why.html
7. A summary of PC Magazine's performance tests can be found in: "Samba runs rings around Win2000," IT Week, April 23, 2002. www.vnunet.com/itweek/news/2084449/samba-runs-rings-win2000
8. Blue Gene/L supercomputer on Linux
9. Framework for Governance in Open Source Communities, csdl2.computer.org/comp/proceedings/hicss/2005/2268/07/22680192a.pdf
10. A Quantitative Profile of a Community of Open Source Linux Developers, www.ibiblio.org/osrt/develpro.html
11. Linux Test Project, www.linuxtestproject.org/
12. CGL Registrations, www.osdl.org/lab_activities/carrier_grade_linux/registration.html/document_view
13. Ferraro-Esparza, V., Gudmandsen, M. and Kristofer Olsson: Ericsson Telecom Server Platform 4. Ericsson Review, Vol. 79(2002):3, pp. 104-113
14. Bjurel, J., Blau, S. and Thyssen, A.: Ericsson's Integrated Site Concept. Ericsson Review, Vol. 82(2005):1, pp. 54-63
15. Holmberg, P. and Isaksson, N.: APZ 212 30—Ericsson's new high-capacity AXE central processor. Ericsson Review, Vol. 76(1999):3, pp. 148-155
16. Free Software Foundation, http://www.fsf.org/
17. Source Erlang, www.erlang.org/