15th Bled Electronic Commerce Conference
eReality: Constructing the eEconomy
Bled, Slovenia, June 17 - 19, 2002

Collaboration in Open Source Environments

Philippe Aigrain

Software Technologies, Information Society, General Directorate, European Commission

This paper was originally prepared for the IPR in ICT-based Collaborations Working Group of European Commission DG Research – drafted 25/03/02 – The present version was revised 13/05/02 for presentation at the BLED Electronic Commerce 2002 Conference.

Disclaimer: Views presented in this paper are only the author's and do not necessarily represent the official position of the European Commission

Abstract

This text provides some background and concrete examples of free / open source software collaborations, and of open information and contents collaborations based on similar licensing schemes. It discussed their impact on dissemination and exploitation of research results. It describes the back offices and front desks of such collaborations. Finally it analyses their needs in terms of IPR and possible conflicts with other approaches.

Acknowledgements and Definitions

This text draws upon the report of the European Working Group on Libre Software, accessible at http://eu.conecta.it/paper.pdf, and the author is very much indebted to the authors of this report for his general understanding of the domain. The reader can refer to this report for an introduction to the history and mechanisms of free / open source software. More recent information is based on the direct implementation of free / open source software policies of the IST programme, but also owes a lot to international activity of the Free / Open Source Software

Research Community (http://opensource.mit.edu) and to the workshops organised by the Center for Information Policy of University of Maryland (http://cip.umd.edu/osagenda.htm) and the Berkeley Roundtable on the International Economy together with the Open Source Development Network (http://www.osdn.com/conferences/brie/BRIE.pdf).

Definitions

A software is **free** (resp. **open source** software) if it is released under a free software (resp. open source software) license. List of approved free software licenses can be found at http://www.fsf.org/licenses/license-list.html and list of approved open source software licenses can be found at http://www.opensource.org/licenses/index.html

All free software or open source software grant users 4 **fundamental freedoms**: to access, use for any purpose, distribute and modify/re-distribute the software. The 4 freedoms constitue an interdependent system : it is together that they enable co-operation and trust. Licenses that remove all or part of one of these freedoms are strongly rejected by the free / open source software developers and movements. The respective players exhibit strong solidarity when their common view of the fundamental freedoms seems threatened. Of course some non-free, non-open source licenses such as the Sun Community Source License used for Java and Jini (not to be mistaken with Sun Public License, which is open source, and is used for instance for JXTA) and the Apple Public Source License, have lead to succesful developments, but without gaining the endorsement of free / open source developers and organisations, and as result in a context of distrust. Microsoft's shared source philosophy grants partially the first freedom and none of the others.

Among free / open source software licenses, the main distinction is with regards to obligations: some licenses make it compulsory to also distribute under the same license any modified software or more generally "derivative work" i.e. software incorporating free software into a set created by direct linking. Such licenses are called **copylefting**. Contrarily to what is often claimed, copylefting licenses make it perfectly possible to run proprietary software "on top of" copylefted software, or the other way round, or to interface for instance through object request brokerage between both. The principal copylefting license is the GNU General Public License (http://www.fsf.org/licenses/gpl.html) which in itself covers more than half of all free or open source software. Non-copylefting licenses allow to put derivative work under proprietary licenses. Some people prefer them because they do not restrict freedom at all, others because of claims that they are more business-friendly, a point which has been subject to much debate. In reality, non-copylefting licenses facilitate the initial transition to open source of proprietary software companies, as they can keep both types of approaches in parallel, but they cancel a critical element of building trust for customers. Some major company projects use the GPL: this is the case for instance for Open.Office (Sun), for many IBM developments, for the RedHat/Cygnus embedded operating system eCos, for the JONAS EJB server (Evidian, formerly Bull), and for the Open.Cascade CAD software (formerly Matra-Datavision). And some important community efforts use non-copylefting licenses, for instance Apache.

Similar licensing schemes have been built for information or contents. Typical examples are the Open Content License (http://opencontent.org/opl.shtml), the Open Publication License (http://opencontent.org/openpub/), and the GNU Free Documentation License (http://www.fsf.org/licenses/licenses.html#FDL).

Though licensing is what defines free / open source software approaches, and most developers consider freedom as a result in itself, licensing is only the enabler for collaborative development, and this collaboration is valuable only if the developed software is useful. In reality, free / open source collaborations constitute a hard reality check on this usefulness. The next sections investigate the substance of free / open source collaborations.

For a more exhaustive set of references to free / open source activities and policies see the links provided in appendix.

Sources and Types of Free / Open Source Software Collaborations

There are 3 main sources of free / open source collaborations: publicly funded infrastructure, grassroots co-operative efforts, and efforts stemming out of business strategies of some industry players. The first 2 types coexisted from the start in the first half of the 1980s, while the third has developed more recently. In practice, one can distinguish more detailed types of collaborations, taking in account also the type of software being developed:

- General information society infrastructure (Internet, Web, grid)

  The Internet Protocol stack software, most essential software for basic Internet applications (bind, sendmail, etc.), World Wide Web software, the computing grid software from Globus and the Global Grid Forum are all free software, and were designed from the start to be so (even if in the case of the Internet it was often before the free software concept was explicited). These developments aim at creating a common information infrastructure. In general they were initiated by relatively small publicly funded teams, but who positioned their work from the start as service to a much larger community of users. The initial design of this software was generally done by a small group of very qualified people, but the later design and development decisions were and are submitted to a wide consultation, under the model of "request for comments" (open to anyone). The choice of free licensing was justified as to obtain maximal dissemination, to ensure trust in supplier independence and to be able to mobilise contributions from all. In practice, these developments amount to the joint production of standards (in arena such as IETF, W3C and GGF) and of their implementations.

- Scientific community software and information resources (astrophysics, biotechnology, high-energy physics, geographical and statistical information, humanities)

  Another major source for free / open source collaborations has been the activity of some scientific communities. These activities arise from distributed communities. The most common case are communities collaborating on instruments (satellites, telescopes, particle accelerators, synchrotrons, etc.), on general databases and catalogs (stellar catalogs, genomic and proteinomic databases, bibliometry, humanities), or developing common tools for the purpose of teaching and research (statistics, robotics, image processing, mathematical software). These projects tend to be collaborative from the start, though their mode of organisation (development model) vary from the very centralised to a pyramidal model. Public funding plays a prominent role, with sometimes an important role of private not-for-profit foundation funding, such as Wellcome Trust funding in the case of genome analysis. The projects focused on instruments are centered on software, while the others have both a software and a free database or information resources dimension. Overall, these projects are creating the infrastructure of a scientific discipline. The case of biotechnology is particularly interesting, as contrarily to high-energy physics or astrophysics, it has great proximity to industry applications. It is one of the domains in which it was possible to observe competition between open public approaches and proprietary private-led efforts, and the open public approach does not fare too bad in the comparison. For a detailed presentation on the subject see: http://2001.istevent.cec.eu.int/library/documents/113.pdf . There is debate on whether this can be generalised or not to other domains, some arguing that only the emotional value of human health made it possible to mobilise sufficient not-for-profit resources. In a similar line, the technology transfer approaches of the National Institute of Health in the US provide interesting distinctions between research instruments (to be licensed under open schemes, whether they are software, databases, cell lines or other entities) and products (possibly appropriated).

- Grassroot efforts: (OS, generic applications, mathematical publishing, peer-to-peer, media technology, artistic platforms)

  This is the most well known and publicised variety of free / open source collaboration. Free software was initiated as a grassroot movement, ethically motivated by the benefits of freedom itself, well before it became a possible business strategy. The core objective was the availability of a free operating system as universal platform for essential information processing activities. This objective took around 10 years to be completed from the creation of the Free Software Foundation to the first truly usable and widely distributed versions of GNU/Linux. By then it was expanded as the notion of a universal platform for information processing encompassing new generic applications (office suites, Web applications, media technology,). In parallel a great number of more specialised efforts developed in domains ranging from scientific modelling and computation to artistic creation. These efforts were precedented by the remarkable achievements of the TEX scientific text layout and typesetting software, instigated by Donald Knuth. Grassroot collaborations are characterised by real distributed development (of mutually co-opted developers), often world wide and round the clock. The development organisation can be more or less structured with a predominance of a flexible pyramidal model. On top sits a core group taking key decisions (for smaller projects, the core group can be limited to one person, and for larger projects there can be a "benevolent dictator" on top of the core group, for instance Linus Torvalds for the Linux kernel). Then there is the wider group of important contributors. Then a large group of contributors of minor elements (patches, bug reports, functionality requests). People move relatively flexibly between these positions, based on recognition by their peers. Getting efficient work done out of such a development process calls for a sound initial design of the software (not always explicited in design documents!), a number of support tools (see section on back offices) and management / human relations skills for the head(s) of the development team. These processes include a very important element of skills building and creativity. To the surprise of the classical software engineering community, they have proven to deliver remarkably reliable software, at least when the user/developer community is wide enough. It is estimated that there are around 250,000 free / open source software developers involved in such efforts. Two thirds of the individual developers are contributing voluntarily without compensation, and one third as part of their job (university, administration or company). Close to 50% of developers spend less than 5 hours a week on free / open source development, and this ability to contribute relatively small efforts plays a key role. Some analysts have highlighted the fact that with such ability to contribute small efforts the issues of incentives and motivation become irrelevant to understand the overall process and impact. 16% of developers spend nonetheless more than 20 hours a week and 7% more than 40 hours a week.

- Mutualised user investment in platforms and inter-operability (embedded platforms, administrations, health systems)

  Many business models have developed around open source software: distribution, support, customised or additional development, etc. One of the most successful ones, illustrated by the case of Cygnus, now part of RedHat, is to develop free / open source software for groups of large users.This may be seen as the industry or organisation consortium approach to the production of common platforms by mutualising investment. Outsourcing it to an external party, and choosing a free (generally copylefting) license guarantees durable supplier (and competitor in the case of industry users) independence. Cygnus is a prominent success by developing an embedded OS which is easily portable to a great variety of processors. This model is present in many different fields, for instance it is today developing for administration software, and in the field of health information systems.

- Business strategies (manufacturers, service companies)

  Open source software is now a major business strategy for a number of industry players. There are 4 mains drivers in this process: the role of manufacturers of hardware, the fact that it can be a natural model for software service companies, the fact that it acts as a corrective to network externalities and increasing returns monopoly effects, and ability of free software to fuel initial usage and build up a potentially interesting market. The case of IBM illustrates the first 2 drivers. IBM is both a manufacturer and the first software service company in the world. As a manufacturer it has made the strategical decision to support Linux for its low cost, reliability and customer acceptance on the server market. As a service company, it is engaged in a major effort of re-engineering based on open source software, including for development tool frameworks such as Eclipse. For service companies, open source software strategies means acceptance of less customer locking in. From this view point, it is a strategy of the strong or of those who wish to conquer new markets. The European software service industry has taken a somewhat conservative position in this respect, position that it might have to review if it is faced with competition of open source offers. SAP has taken some initial steps in that direction. Some software publishers have recently developed visible strategies for products losing market share or confined in secondary position, by making them open source and re-engineering their activity around services. Typical examples are Netscape with Mozilla, Matra-Datavision with OpenCascade, Bull/Evidian with its versions of EJB servers, Sun with Open.Office. Such approaches call for a very important initial investment: a product which was not originally designed to be open source has to be re-engineered and strong promotion is necessary to build a community of developers around it. Some specialised companies (CollabNet for instance) make a special business of supporting such efforts. Finally, open source software plays a key role in the development of new personal usage and resulting markets, in particular when initially they are not clearly solvable. They are often used by companies to develop new business in areas that do not constitute their core activity, a typical example being the developer community created by Nokia for its Media terminal set-top box and home gateway project.

Impact on Dissemination and Exploitation of Software Research Results

In a field like software, innovation paths are somewhat complex. Contrary to what happens for physical devices technology, ideas are easily transported from one domain to a completely different one. Components can be combined at large scale without quickly resulting in severe physical constraints or bottlenecks. In addition, software innovation dissemination is most of the times dependent on an environment of human usage which make its domain of usage very hard to predict initially. It is coupled with and constrained by factors that can be sociological, psychologial, economical, cultural or dependent on other technology environements to a degree much greater than for tangible technology is. Instant messaging was invented for businessmen, but its present success whether on mobile phones or on PCs rests on teenagers groups. JPEG (discrete cosine transform based image compression) was invented in the 1970s for satellite imagery transmission, but it is the basis for a much bigger industry of digital photography. A side effect of all this is that software innovation, in those cases where one can trace it to specific sources, is rarely exploited by those companies or organisations that originated it. The example of Windows-Icons-Menu-Pointer user interfaces (WIMP, that constitute the basis of today's user interfaces) originated in SRI and Xerox PARC, never successfully exploited by Xerox, in particular because it was too focussed on the office environment that provided the initial inspiration for this innovative metaphor, copied and re-engineered by Apple for personal users, finally copied by Windows is typical. Nonetheless, software innovation is strongly embedded in know-how and people. Ideas travel from one domain to another, but they often travel with people.

In such a context, it is not surprising that there is lively debate on what are the best mechanisms for favouring dissemination and exploitation of software innovation. The general trend observed from the early 1980s of trying to organise dissemination of research results (in particular for universities) by creating strong intellectual property, and licensing it, or creating spin-offs has been applied to software just as to other fields. In parallel, and in exactly the same period, the European RTD programmes have developed, consistently with their 50% funding scheme, with IPR / access rights rules centered on ownerships of results by their originators, and relatively restrictive access rights. Nobody can presently claim to have strong quantified evidence to measure the general dissemination and exploitation results of such approaches in comparison to more open licensing schemes. I have personally developed an analysis that tends to showing that in the special case of algorithmic software patents, there is a negative effect of strong property rights on progress and dissimination of innovation. See http://cip.umd.edu/Aigrain.htm#Q5. But this is far from being any real proof, or fit for generalisation to other forms of property rights. More serious modelling by Bessen and Maskin at MIT gave consistent indications, and insight on the reasons for such findings (see: http://www.researchoninnovation.org/patent.pdf). On the contrary, there is more than anecdotal evidence that open licensing was a key ingredient of some of the greatest successes of software innovation dissemination. In additions to the examples already given in the section on source and types of open source collaborations, one can mention again JPEG compression, for which the wide availability of a free software implementation of the coder (the independent JPEG group coder) played a key role in developing new applications, some of which account for huge markets.

Let's thus just say that there is at least enough ground to justify keeping wide open the option of free licensing for dissemination of publicly funded research results, in particular those from academic research. The UK government has recently published a consultation (http://www.govtalk.gov.uk/documents/OSS%20Policy%20draft%20for%20public%20consultation.doc) on the option of making open source licenses the default dissemination mechanism for university research results. Many scientists have initiated (notably in the biotechnology research field) grassroot actions to ask for public research software results ... to be public software. See for instance the Open Informatics Petition (http://www.openinformatics.org/petition.html).

Synergy between Free / Open Source Software and Open Information and Contents

As already shown in the case of scientific communities, free / open source software often goes in association with freely licensed information resources. There are deep reasons for this. The first one is that the border between software and information is more and more blurred: is a script in a Web page one or the other? More generally, some types of information or contents have properties that make their production or usage benefit from the "no transaction cost" co-operation at large scale[1]. This is the case for entities that can be produced by relatively small (in terms of efforts or investment) incremental steps, or whose quality benefits from extensive usage or criticism, or for which the production can be easily be distributed between large numbers of people. Typical examples are information fora, databases with distributed production, textual and photographical databases, bibliography, educational contents, etc. Despite this, there is great uncertainty on the range of possible development for open contents, mostly because the economical activities that can be coupled with the existence of an infrastructure of open contents are not as clear yet as for software.

Open licensing for information and contents is presently strong only in a limited number of segments: infrastructural scientific databases, cultural heritage and public domain documents, digital photographic exchange, educational resources, scientific publishing and co-operative media (information fora coming from the Usenet tradition, co-operative media such as Slashdot or Indymedia). But in these domains, it has taken an irresistible momentum. It is supported by ethical and practical motivation of the creators themselves, who in those case were already

[1] A classical reasoning (called Coasian analysis in technology economics) led in the past to consider as positive the attribution of property rights on innovation resources, provided that there is a sufficiently fluid market for access to use rights. This analysis breaks down when there is competition with innovation approaches based on quasi-null transaction costs such as those permitted by global ICT, and when innovation steps do not require major upfront investments.

supported by indirect funding, and received only very limited reward (in average) from property rights. They basically have nothing to lose and a lot to gain from sharing. This process is still met with mixed views by some of their funding agencies (for instance universities for teachers), but since the supposed revenues from commercialisation by public / private partnerships of publicly funded information have proved to be very elusive, the trend is clearly moving towards accepting free dissemination. This is also reflected in recent regulatory initiatives such as for access to public sector information. The organisation and back office of open information / contents projects are very similar to those of free / open source software, are based on free software, but exhibit also specific innovation (for example distributed annotation and peer-to-peer publishing).

The Back Office and Front Desk of Free / Open Source Collaborations

Open source collaborations depends on critical production back-office and no less critical dissemination front desks. Due to the continuum of developer / user positions, the 2 must be organised in a consistent cycle. In free / open source software, the critical components of the production back office are project hosting, versioning tools, and bug tracking. On the front desk, packaging for distribution, dependency analysis, package and component search, rating and advice systems play the key role. **Project hosting** consists in hosting the code base and associated documentation, and production of builds (compiled code), together with associated tools and security layers. Presently project hosting for a given project is predominantly centralised. There are very large project repositories such as SourceForge for the Open Source Development Network (http://sfads.osdn.com/1.html), which hosts 20000+ projects with 160000+ registered developers or Savannah (http://savannah.gnu.org/) for the GNU project of the Free Software Foundation. More specific projects, in particular those sponsored by companies, use specific hosting facilities, either directly hosted by the organisations or subcontracted to a specialised company like CollabNet. More recently, agencies have been created as hosting portals for development in specific domains, such as the German public-private agency BERLIOS (www.berlios.de) for administration software, or projects in the works for similar portals at local, regional or European level.

A key aspect of open source collaboration is that distributed developers must be able to work in parallel on different but possibly interdependent parts of the the code base for a project, and it must be possible to merge, adapt and test the results. The enabling tools are called **versioning tools**, and they are associated with **configuration management** tools (to be able to address a great variety of target environments) and more recently automated build tools. The key versioning tool is CVS, almost synonym of free / open source development. Recently people have been trying to overcome some of the complexity or limitations of CVS by developing better user interfaces, having a better support for branch and merges, supporting better internationalisation and multilingual versions, and going into the direction of peer-to-peer distributed hosting. The relevant projects in that respect are Subversion from CollabNet/Tigris (www.tigris.org) and arch – name to be changed – at http://www.regexps.com/#arch. Some large scale projects such as Mozilla (the open source Netscape browser effort), that involve a very great number of developers, have developed tools for creating and testing automatically daily builds of the software being developed. These tools represent a true "industrialisation" of the open source collaboration.

Design documentation is traditionally not the strongest point of open source software. Projects like ArgoUML (http://argouml.tigris.org/, more generally the full design chain of Tigris at www.tigris.org) try to overcome this weakness by providing better design and documentation support. This is an important problem in view of ability to redesign open source code when, through its development, it reaches a stage where such redesign is needed (for instance because of changing requirements or usage).

In addition to code production, **bug reporting and tracking** is a key element, notably because it put users in the loop. Bugzilla (www.bugzilla.org), a tool developed in the frame of the Mozilla project is the reference tool. To give an idea of the scale of the problems, there are 180,000+ bugs tracked for Mozilla (counting solved ones), and this is not because it is particularly buggy, but because bugs get tracked in open source projects. In addition, functionality requests are also important for putting non-developing users in the loop. It is achieved using a variety of simple Web-based information systems.

On the front desk, packaging, promotion, search tools, rating systems are no less important. Project hosts provide access to the source code, and generally to a limited set of binary distributions adapted to specific execution environements. The main distribution channels are though distributors. **Packaging** has become a difficult problem with the multiplication of distributors, often using different package formats. There is on-going work to try to come to a simpler set of agreed packaging. The key difficulty in packaging and distribution is solving the dependencies between various packages. Distributors provide stable consistent sets of packages, but when one wants to install a specific tool without waiting for its inclusion in a distribution, one runs often into dependency problems. This is problem common with proprietary software, but intensified by the great variety of development projects. Open source developers try to keep it under control by sticking to relatively self-contained modular packages and by providing support tools solving dependencies, but it is a hard challenge.

In parallel, seen from the user or developer side comes the issue of **locating the relevant package** for a given task, or components of code that can be re-used in a given context. Sites like Freshmeat (http://freshmeat.net/) play a key role in this process. Innovative tools are being developed for open code indexing, searching and reuse, including with EC research funding support.

For open contents, additional tools are distributed annotation (allowing contributors to annotate segments of a document or databases, and users to view the annotations independently of their source), threading (for information fora) and information space views, peer-to-peer publishing, and collaborative rating.

IPR Needs and Dangers for Free / Open Source Collaborations

Free and open source software approaches are based on intellectual property, notably copyright, as the foundation entitling those who own it to license under a free / open license. But this is clearly an atypical (for intellectual property specialists) usage of intellectual property, particularly in its copylefting version, in which a property right is used to guarantee that it is not possible to appropriate the entity which is the object of the right.

In practice, the compatibility of intellectual property modalities, mechanisms and management with free / open source approaches depends mostly on transaction costs issues. **Copyright** raises no problem in itself, because it is granted without transaction, can be licensed freely when desired without entering into contracts, and does not cover underlying ideas but only expression forms. **Copyright protection technology** raises many concerns, in particular when it includes aspects regarding **outlawing of circumvention**, and specially if inter-operability is not considered as a valid reason for circumventing, or accepted only when there is significant commercial[2] usage. More generally efforts to hard code (in technology) features that are justified only to implement some particular strategies of digital rights management have side effects that can represent serious dangers for free / open source software. For instance, if one tries to make it impossible to circumvent technology measures for copyright protection by closing hardware and OS booting, it can make dual booting of operating systems or runnning open source applications on top of propietary OSes impossible. As these are key transition paths for users between proprietary and open source software, the effect would be strongly anti-competitive.

Trademarks are generally considered as a valid intellectual property scheme by free/open source software developers. There have been some litigations (for instance on the trademark SSH), but they were easily solved, and other examples such as Linux (trademark of Linus Torvalds) or OpenCascade, show that it is very well possible to use trademarks to protect free/open source software against proprietary counterfeiting, or for branding a specific business version of free / open source software.

[2] Interestingly, the English and French versions of the 2001/29/CE Copyright in the Information Society Directive contain contradictory wording on this clause!

Software patents are seen as a major danger for free / open source software. Claims to the contrary are sometimes heard. Intellectual property specialists claim that open source software inventors could on the contrary use patents to protect their approaches against proprietary competitors. Such affirmations ignore the concrete mechanisms and motivation of open source development. Open source innovators would not even consider other than for a joke, or in very special cases, any scheme to gain a property right that involves initial transaction costs of the size of those associated with patents, and a delay of 2-3 years minimum. On the contrary, the potential litigation threats associated with the existence of hundred of thousands of software patents, each of which can cover some of hundreds of features or components ideas of a developed software are extremely high. Community developments manage often to simply ignore them, and exert image pressure on companies if they start litigating. But more institutionalised processes, such as standards reference implementations are now daily stalled by software patent litigations threats. There are major blockage in IETF (on domain name localization), in W3C (for the RDF standard), in ETSI, in MPEG-4 (on terms of licensing). W3C recently confirmed, after severe public criticism, that it would no longer consider accepting RAND (Reasonable and Non-Discriminatory) licensing for software patents related to standards, but keep requiring royalty-free licensing.

Licensing rules, for instance those associate with research funding IPR rules have a strong impact on the practical feasibility of open licensing dissemination. Key issues are those of requirements on intellectual property protection, granting of access rights including rights to sublicense, and more importantly, at which stage does one agree on a licensing scheme. If it is left for consortia to agree, and at an undefinite time on what scheme they will choose, then the likelihood that one of the partners will block open licensing becomes extremely high (for reasons explained in the section on exploitation and dissemination paths). Depending on whether it is possible for a research programme to specify in call for proposals preferred dissemination schemes for a given activity, or whether it is possible to require choices to be precised in proposals, or if they can be negotiated prior to a funding contract, or finally if they are left to a later agreement, the practical feasibility of open source dissemination routes will be completely different.

About the Author

Philippe Aigrain is Head of Sector "Software Technologies" in the unit "Technologies and Engineering for Software, Systems and Services" of the European Commission Information Society Technologies R&D Programme, in which he is in charge of actions in support to free / open source software and related innovation. He was trained as a mathematician and theoretical computer scientist, and holds a Doctorat and the Habilitation à Diriger les Recherches from University Paris 7. From 1972 to 1981, he worked in software engineering research groups of

software companies. He was a research fellow at U.C. Berkeley in 1982. Since then, and before joining the European Commission in 1996, he headed research teams in the field of computer processing, indexing, retrieval and interaction for audiovisual media (video, music, still images). He his the author of more than 60 technical papers, as well as of papers on the history, economy and sociology of information exchanges.

Appendix: Free / Open Source Projects and Policies

Specific research funding:

- IST programme actions: http://www.cordis.lu/ist/ka4/tesss/impl_free.htm
- French “Réseau National des Technologies Logicielles”: http://www.industrie.gouv.fr/rntl/Rapport-GTLL-RNTLfinal.pdf
- South-African action plan: http://www.naci.org.za/docs/opensource.html
- EUROLATIS Latin America / Europe co-operation: http://www.eurolatis.upm.es/eurolatis.asp?MN_menu=MN_ws7

Mainstreaming of the scheme for publicly funded research:

- UK policy: http://www.govtalk.gov.uk/documents/OSS%20Policy%20draft%20for%20public%20consultation.pdf
- OECD Global Research Village Conference on Access to Publicly Funded Research: http://www.minocw.nl/english/conferentie/deel2.pdf
- US National Academies CSTB: http://www4.nationalacademies.org/cpsma/cstb.nsf/web/project_opensource_prospectus?OpenDocument
- Open Informatics Petition: http://www.openinformatics.org/petition.html
- Large research initiatives adopting an open source approach directly: see Global Grid Forum (www.gridforum.org), ENSEMBL (www.ensembl.org) initiative in bio-technology, etc.

European industry initiatives

- Object Web consortium: www.objectweb.org
- Open Cascade (www.opencascade.com, www.opencascade.org)
- Nokia Media Terminal: http://www.nokia.com/multimedia/pdf/developers_brochure.pdf
- Thalès and Alcatel participation to the open source software approaches to security RIS network

- SAP database software release

E-government policies, adoption in European and National administrations:

- E-Europe 2002 action plan education and e-government actions: http://europa.eu.int/information_society/eeurope/action_plan/pdf/actionplan_en.pdf
- IDA programme: http://europa.eu.int/ISPO/ida/
- Limited approach (servers) to introduction in Commission's informatics
- European Parliament resolution of 5 September 2001: http://www3.europarl.eu.int/omk/omisapir.so/pv2?PRG=QUERY&APP=PV2&LANGUE=EN&TYPEF=A5&FILE=BIBLIO01&NUMERO=0264&YEAR=01
- Initiatives for introduction in National parliaments, see for instance: http://www.bundestux.de/english.html
- German KBST (http://linux.kbst.bund.de/) and BMWi Sicherheit-im-Internet (https://www.sicherheit-im-internet.de/themes/themes.phtml?ttid=2).
- French ATICA: http://www.atica.pm.gouv.fr/bouquet-libre/
- South-African action plan: http://www.naci.org.za/docs/opensource.html
- Information society programmes actively supporting free software adoption in administrations in Canada, Brazil, Peru, Venezuela

Direct provisioning:

- German GnuPG and Ägypten initiatives (https://www.sicherheit-im-internet.de/themes/themes.phtml?ttid=86, http://www.gnupg.org/aegypten/)

Local government

- France: http://www.illico.org/public/dossiers/obstll/, http://www.soissons-technopole.org/
- German Länder policy discussion, see: http://www.heise.de/newsticker/data/odi-22.01.02-001/
- Spain: region of Extramadura: http://2001.istevent.cec.eu.int/library/documents/52.pdf
- Brazil: legislation and policies in states of Rio Grande del Sul and Santa Catarina

Legislative proposals and political think tank statements

- French Le Déault-Paul-Cohen proposal: www.osslaw.org and previous Laffitte-Tregouet proposal
- Belgian law proposal: http://www.lachambre.be/documents/1022/1.pdf

- Danish law proposal: http://www.folketinget.dk/Samling/19991/udvbilag/FKU/B114_bilagOS.htm
- Fondation Jean Jaurès: "Vers la Cité Numérique: Un projet politique pour la société de l'information". See also related French Socialist Party policy statement: http://www.parti-socialiste.fr/tic/ps-tic_2002.php
- Law proposal in Peru

Agencies:

- French ATICA (see above) and RNTL proposal for a National Foundation (see above)
- German BERLIOS (www.berlios.de)
- Benkler's US proposal: http://www.law.nyu.edu/benklery/WhitePaper.pdf
- IDA portal for administration open source software (in study phase)
- Intranet Métropolitain de la Communauté Urbaine de Lille (study phase, http://www.artesi-idf.com/ville/services.php?sId=118)

Intellectual property reform and "commons" analysis

- Many analysts have highlighted that classical intellectual property thinking is poorly suited to handle intangible information entities, stressed the dangers of its recent dramatic extension, or proposed alternative frameworks. The key references are Lawrence Lessig, Yochai Benkler, John Perry Barlow and more generally EFF, Jamie Love and CPT, Paul Starr, James Boyle, and Richard Stallman in the US, Jean-Claude Guédon in Canada, Philippe Quéau (UNESCO), Bernard Benhamou and the author of this text in Europe.

Social and economic research

- Actions in the IST programme (see for instance FLOSS: http://www.infonomics.nl/FLOSS/)
- Global agenda setting (http://cip.umd.edu/osagenda.htm)
- The BCG survey: http://www.osdn.com/bcg/bcg/bcghackersurvey.html

Open information and open contents initiatives

- Project Gutenberg: http://promo.net/pg/
- Nupedia: http://www.nupedia.org/main.shtml
- ENSEMBL (see above)
- Public Library of Science (http://www.publiclibraryofscience.org/), scientific publishing repositories at Los Alamos, Cornell etc (http://www.arxiv.org/, http://lib-www.lanl.gov/lww/welcome.html, http://www.openarchives.org/)

- Eprints initiative (supported by the Soros Foundation): www.eprints.org, Public Archives of the Humanities project (hypernietzsche.org)
- Copylefted art: http://artlibre.org/
- Specialised (ex: slashdot.org) and alternative media (ex: www.indymedia.org)
- See also www.opencontents.org