http://www.irill.org

Initiative pour la Recherche et l'Innovation sur le Logiciel Libre

Research and Innovation on Free Software

Roberto Di Cosmo

Fossa, Grenoble, November 8th 2010

Context : FOSS = radical change

Society :

- Contributes to scientific and technological commons;
- Give back to citizens control of strategic technologies

Technology :

- Changes how software is designed, developed, distributed, sold, maintained, customised and taught; from generic infrastructure to specialised applications; in more traditional areas, and in the emerging embedded and mobile market

Industry/Economy

- Reopens to competition entire sectors of the Economy
- Reduces time to market with software built faster and cheaper
- Reduces cost, and fosters vendor independence in all sectors of activity

Accompanying FOSS growth

Face the new challenges

- scientifc,
  - New scientific and technological problems are brought to the limelight from FOSS, with an unprecedented change of scale in software artifact size and speed of evolution,
- educational,
  - Teach tomorrow's engineers how to develop software in the FOSS environment, and help them stay up to date,
  - Educate technical and non technical newcomers to FOSS values and principles, and include them in the community
- economic,
  - Help transfer from academia to industry, and ease bottom-up innovation
  - Foster an open ecosystem approach to information technology

See also INRIA's strategic plan

http://www.inria.fr/valorisation/logiciel%20libre.fr.html

IRILL's goals and ambition

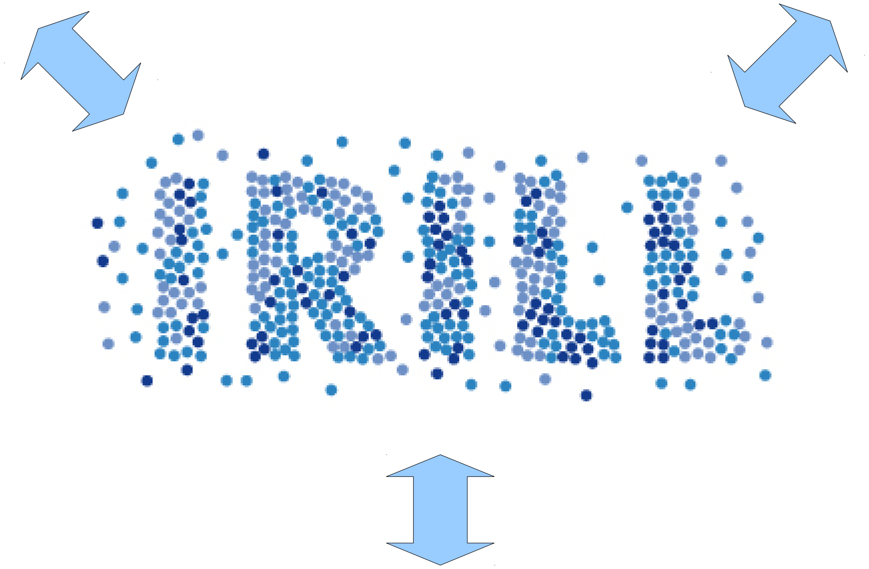
Contribute to all three challenges, and become an internationally recognized center on FOSS development, research and education.

Three main axes:

- A reference center for safe and reliable FOSS
  - Work on the technological breakthroughs our enterprises need
- A reference center for developing new curricula for CS with FLOSS
  - Help educating the qualified engineers we all need
- An experimentation platform for FLOSS technology transfer

IRILL : a place to bind them all

Three communities meet: academia, developers and industry



Developer Communities

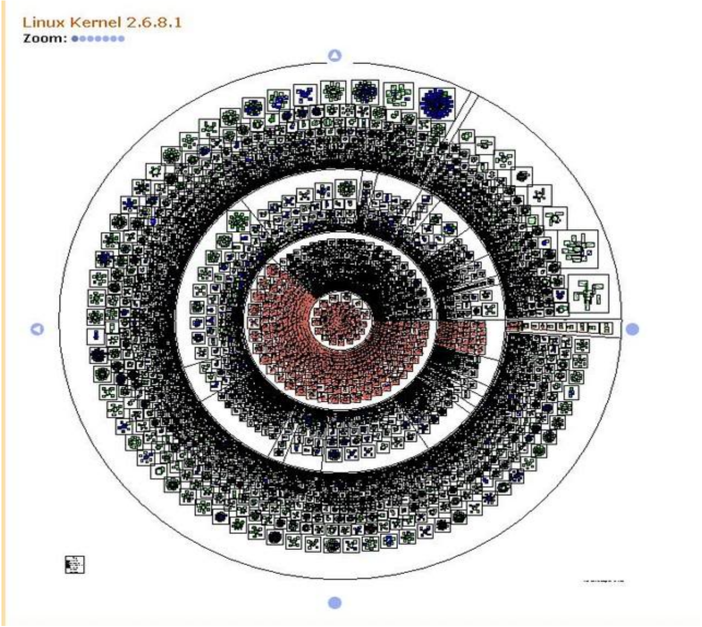
Industries

Research and Education

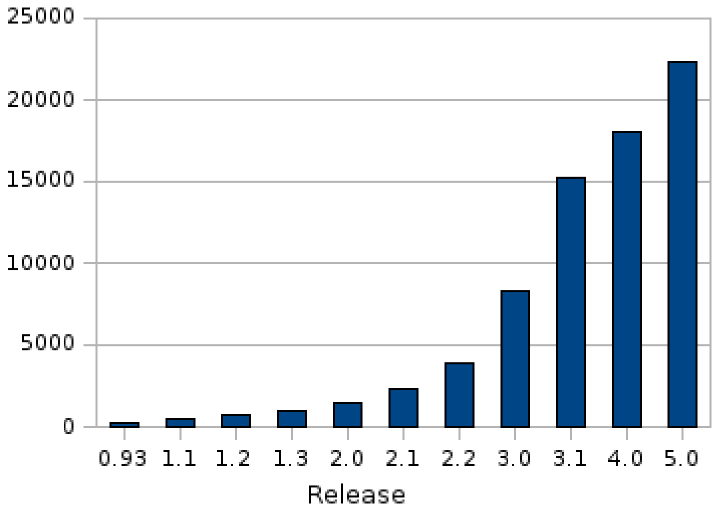
RESEARCH

Change of scale

Component size



Component number

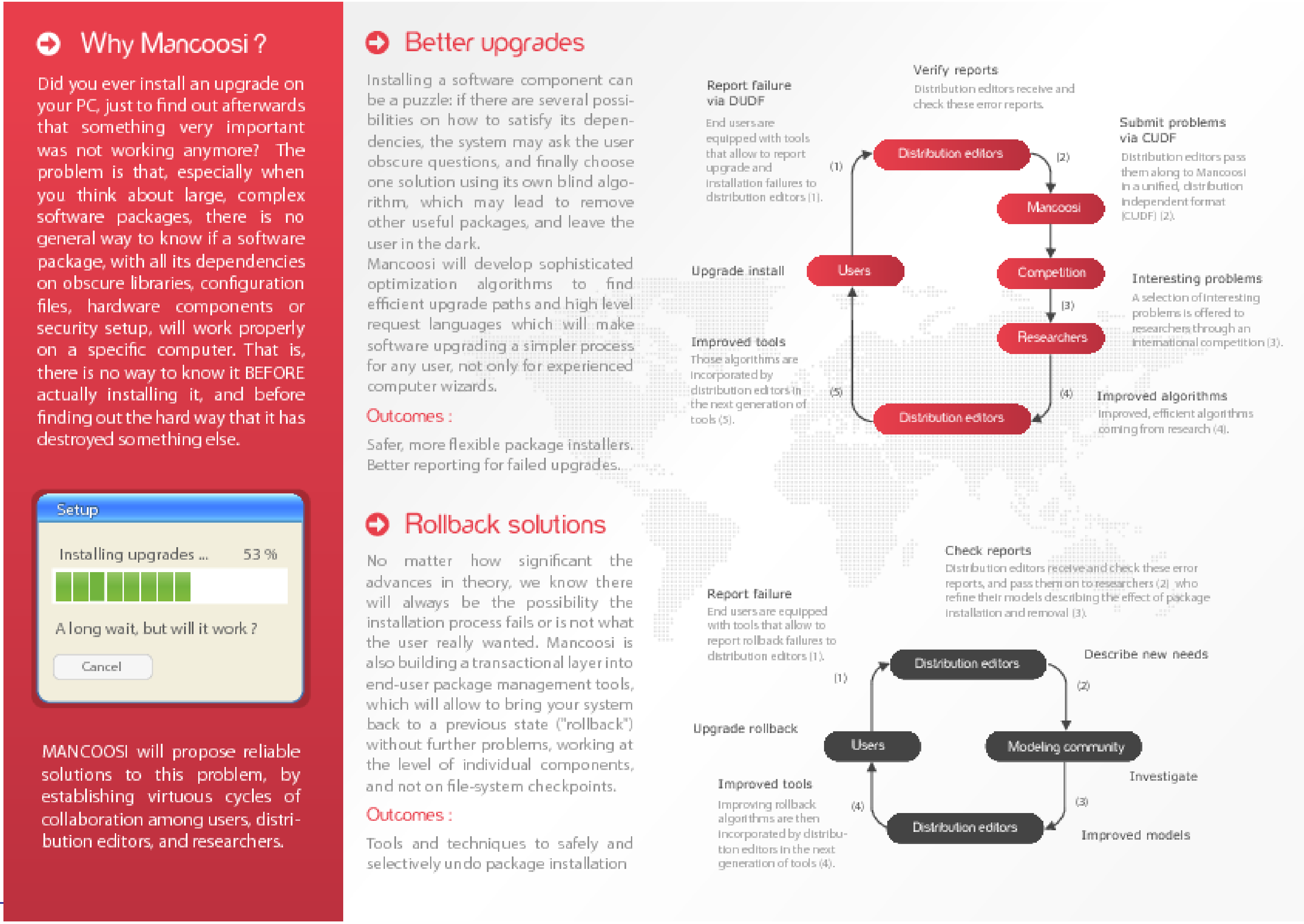


Over 8 M SLOC

Over 20.000 packages

We need measures and tools to scale up

Open innovation virtous cycles : the Mancoosi example



Why Mancoosi ?

Did you ever install an upgrade on your PC, just to find out afterwards that something very important was not working anymore? The problem is that, especially when you think about large, complex software packages, there is no general way to know if a software package, with all its dependencies on obscure libraries, configuration files, hardware components or security setup, will work properly on a specific computer. That is, there is no way to know it BEFORE actually installing it, and before finding out the hard way that it has destroyed something else.

Setup

Installing upgrades ... 53 %

A long wait, but will it work ?

MANCOOSI will propose reliable solutions to this problem, by establishing virtuous cycles of collaboration among users, distribution editors, and researchers.

Better upgrades

Installing a software component can be a puzzle: if there are several possibilities on how to satisfy its dependencies, the system may ask the user obscure questions, and finally choose one solution using its own blind algorithm, which may lead to remove other useful packages, and leave the user in the dark.

Mancoosi will develop sophisticated optimization algorithms to find efficient upgrade paths and high level request languages which will make software upgrading a simpler process for any user, not only for experienced computer wizards.

Outcomes :

Safer, more flexible package installers. Better reporting for failed upgrades.

Report failure via DUDF

End users are equipped with tools that allow to report upgrade and installation failures to distribution editors (1).

Verify reports

Distribution editors receive and check these error reports.

Submit problems via CUDF

Distribution editors pass them along to Mancoosi in a unified, distribution independent format (CUDF) (2).

Upgrade install

Interesting problems

A selection of interesting problems is offered to researchers through an international competition (3).

Improved tools

Those algorithms are incorporated by distribution editors in the next generation of tools (5).

Improved algorithms

Improved, efficient algorithms coming from research (4).

Rollback solutions

No matter how significant the advances in theory, we know there will always be the possibility the installation process fails or is not what the user really wanted. Mancoosi is also building a transactional layer into end-user package management tools, which will allow to bring your system back to a previous state ("rollback") without further problems, working at the level of individual components, and not on file-system checkpoints.

Outcomes :

Tools and techniques to safely and selectively undo package installation

Check reports

Distribution editors receive and check these error reports, and pass them on to researchers (2) who refine their models describing the effect of package installation and removal (3).

Report failure

End users are equipped with tools that allow to report rollback failures to distribution editors (1).

Describe new needs

Upgrade rollback

Investigate

Improved tools

Improving rollback algorithms are then incorporated by distribution editors in the next generation of tools (4).

Improved models

EDUCATION

Education to FOSS, education with FOSS

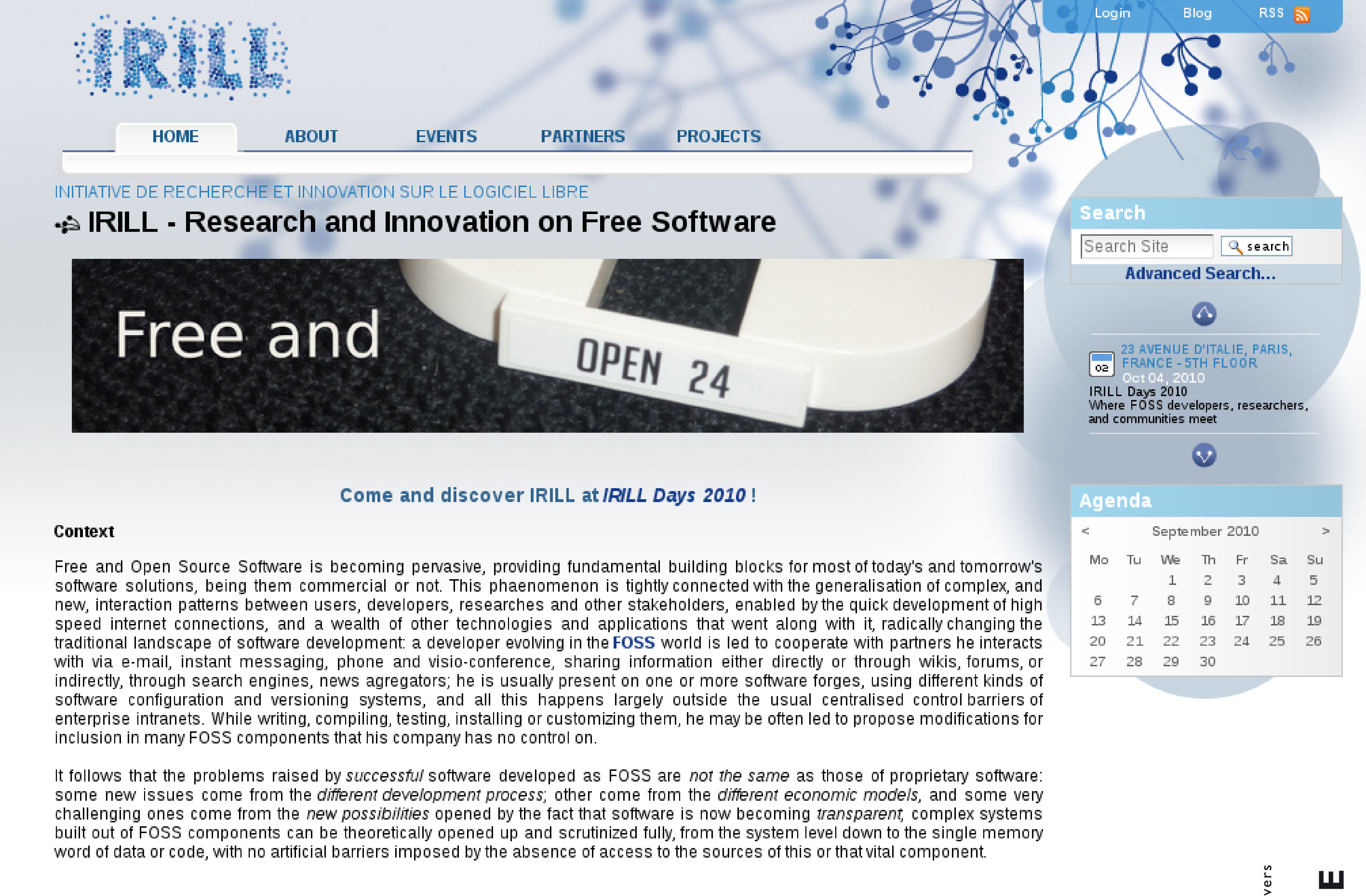
Patterson, Mars 2006, Communications of the ACM :

JOIN THE OPEN SOURCE MOVEMENT.

> Most schools teach "write programs from blank sheet of paper" programming, of which there is very little real-world bearing. A different approach is to leverage high-quality examples of the open source movement. [. . .] Do civil engineering students get to contribute to the construction of a real bridge in the classroom ? [. . .] The recruiting pitch is to join CS and learn in part by contributing immediately to the real world.

The challenge

- Increasing IT proficiency and education early in education, for everybody
  - See Gilles Dowek's talk
- Introducing FOSS and collaboration skills in academia for all : *educating users, enabling them to make a difference*
  - See Ralf Treinen's talk
- Introducing FOSS in CS Curricula
  - See the reports from FTA and Evry
- Adapting CS Curricula
  - See the effort started at http://oscurr.v2.cs.unibo.it/
- Making connections between communities, industries and academy
  - What Fossa is all about
- Building the education and collaboration infrastructure (textbooks, forges, coordination)
- Recognize and reward quality efforts in Academia in this direction



INITIATIVE DE RECHERCHE ET INNOVATION SUR LE LOGICIEL LIBRE

IRILL - Research and Innovation on Free Software

Come and discover IRILL at *IRILL Days 2010* !

Context

Free and Open Source Software is becoming pervasive, providing fundamental building blocks for most of today's and tomorrow's software solutions, being them commercial or not. This phaenomenon is tightly connected with the generalisation of complex, and new, interaction patterns between users, developers, researches and other stakeholders, enabled by the quick development of high speed internet connections, and a wealth of other technologies and applications that went along with it, radically changing the traditional landscape of software development: a developer evolving in the **FOSS** world is led to cooperate with partners he interacts with via e-mail, instant messaging, phone and visio-conference, sharing information either directly or through wikis, forums, or indirectly, through search engines, news agregators; he is usually present on one or more software forges, using different kinds of software configuration and versioning systems, and all this happens largely outside the usual centralised control barriers of enterprise intranets. While writing, compiling, testing, installing or customizing them, he may be often led to propose modifications for inclusion in many FOSS components that his company has no control on.

It follows that the problems raised by *successful* software developed as FOSS are *not the same* as those of proprietary software: some new issues come from the *different development process*; other come from the *different economic models*, and some very challenging ones come from the *new possibilities* opened by the fact that software is now becoming *transparent*, complex systems built out of FOSS components can be theoretically opened up and scrutinized fully, from the system level down to the single memory word of data or code, with no artificial barriers imposed by the absence of access to the sources of this or that vital component.

Search

Search Site [search]

Advanced Search...

02
23 AVENUE D'ITALIE, PARIS, FRANCE - 5TH FLOOR
Oct 04, 2010
IRILL Days 2010
Where FOSS developers, researchers, and communities meet

Agenda

< September 2010 >

Mo	Tu	We	Th	Fr	Sa	Su
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			