Free Software as a Commons – Lyon 27 Oct. 2011
Arturo Di Corinto

Software is not a ordinary object but an instrument good at building other tools and instruments.

Why software is so important? Like writing is an intermediary between us and the others, software is the intermediary between us and all every day's activities.

Software instructs computers in writing letters, calculating, drawing, and it's also useful for writing music, recording stories and ideas and projecting machines and environments, for recording climate changes or warning us of an earthquake.

Today, most part of our communication - and its democratization - relies on software, and since communication is a universal right, a basic human need, and the foundation of all social organization, we must be aware of the role of software in today's communication.

Software is a peculiar language form through which people manage culture and knowledge accumulated over centuries and because of that it's not a simple utensil.

Software is a cognitive artifact and it incorporates intelligence and work, it transmits meanings and values, it brings with it the idea of its creator and of its user.

The fact that its use is exclusive, limited in space and time, or vice versa that it might be modified, given, freely exchanged behind geographical barriers, makes a remarkable difference.

If language is the "operating system" of our society, software is the language of all new forms of immaterial production, and of scientific and technological innovation in the information society. Because of this it has to be free, that is freely usable, to favour the progress of all the society, of every society, because, otherwise, we would be all less free.

Think if someone patented natural languages. A comic strip story of 1991, made by Zzywwuruth e Cicare' and published in Italy by Editori del Grifo, is really prophetic over this terrible perspective.

The "bad" in the story, a certain Brevetto Protoplasto, ask the courthouse to have his rights over the dictionary and the kitchen granted, provoking hilarity among the court and the audience.

The judge states that in order to address such request he should at least demonstrate that he has invented the fire and the word. The bad claims them as Adams's descendant.
But, according to the judge, we are all Adam's descendants and we have all equal rights of talking and cooking. But Brevetto claims that he is Adam's direct descendant by means of Caino and that, in conflict with other human beings, he can prove it: his proofs are a precise family tree compiled and countersigned by priests, bonzos, and muezzins, even by Darwin, and particularly he has the apple of the Adam's original sin brought intact after 7000 generations.

Having examinated the proofs the court is forced to acknowledge to him the ownership of every language and cook recipes and since that moment humans are forced to pay a toll to Brevetto Protoplasto everytime a word is spoken or someone cooks something, actually making the humanity slave as insolvent debtor towards Brevetto Protoplasto.

If the language would be patented thus we would be suddenly slaves, forced to pay a toll everytime that we “execute it” and copyright on cook recepies would prevent even our grand mother to give them to us if we should'nt pay royalties to its inventor.

Something similar has already occured for software.

Fourty years ago software was copyright free, but someone, convinced that profits could'nt be made without applying a user licence, took possession of it putting a label and setting such and many restrictions to its use that today is impossible using it for studing or research scopes or for a scope not in asset with the license without paying for it.

Copyright on software has interrupted the cooperative practice of decentralized planning**,** collective review, free exchange which have characterized its birth and initial development**,** limiting the three main ways in which the accumulation of human knowledge proceeds and which is well explained by public scientific research.

As time passed by copyrighted software has been imposed to the most part of the users by means of unscrupulous business operations or aggressive market campaigns, with the complicity of who, the public decision makers, did'nt understand the potential that such transformation was bringing, but even as the result of more prosaic lobbying operations which did'nt allowed an independent estimation of such effects.

Thus the most important software owner of the world has built an empire through a monopoly which excluded others from the advantages that software was bringing in terms of social and economic progress.

But as long as all those facts occurred a new fascinating adventure began: the battle for the liberation of software.

An extraordinary civilization struggle focused on bringing back into collective dominion what it was public at the beginnings, computer's instruction language, that is software.
Giving back software's ownership to his creators and enlarging user's rights was the first objective of this struggle.

Nowadays every time we pay for a proprietary software subjected to exclusive use license with no source code included we are not owners but licensees and the personal use of it is tied up by the restrictions enclosed with the license which often we don't fully read.

It happens indeed that if software does'nt work properly we cannot open it to get into the mechanism in

order to mend its bugs. If we would like to study its characteristics to improve it, it is not allowed, if we would like to copy it on another of our computers we cannot do it and if we borrow it or if we want to give it as a present to a friend we become criminals. Ignorance is not admitted by the law.

What is obstructed is the aesthetic and functional reinterpretation of software, the same customs that, talking about every day's life tools, are the basis of social creativity and innovation which proceeds indeed through the ability of approaching an issue from a different and unusual perspective.

Moreover this practice which makes us all criminals for the law keeps from building and enhancing that social tie which come by means of free and unlimited exchange, discoveries and intuitions, and so it makes us every time less joint and a little less free.

A mechanism which could prevent all this has been generated by the birth of the GNU project and by the Free Software Foundation and subsequently with the starting of the Open Source Initiative.

A user license, which even if based on copyright overturns it and transforms it in "author's permission", has been written taking the copyright back to the bed in which was born, the one of the need for protecting the written work's authors from the possible appropriation by the publishers and at the same time the need for yielding necessary earnings to the authors and to the publishers to continue this praiseworthy work of knowledge spreading with balanced rights.

This license is the GPL, the General Public License (www.gnu.org), which draw on and enlarge the original concept of fair use for copyright, that is the fair usage of an intellectual work, which allows "protecting the wine instead of the bottle", pledging profits to creators without damaging the users which, thanks to it, buy the right to improve the product, to exchange it, to enhance it, restarting up the virtuous circle of innovation.

This license and the enthusiastic consent to free software philosophy of thousands of developers has generated a big innovation: the GNU/Linux operating system and many others auxiliaries free software applications good also at replacing proprietary software on less expert user's computers.

Behind this grand philosophy is the idea that software have to be a *Commons:* a common good, universal, not exclusive and inalienable. But behind the implementation of GPL and of its derivatives there's a mechanism which prevents, differently from the Commons epoch, the temporary advantage of a selfish behaviour, the individual appropriation of part of the common good, which makes at the beginning richer the individual but in the long run makes everybody poorer, to be useless and self-defeating, asserting the idea that, in an interdependent society, individual welfare is strictly tighten with the collective one.

Remarkable topic for those who feel averse to the logics of profit which proceed to the appropriation of what is a collective property. By our side we are not afraid to say that the privatization of common knowledge is a rip-off. On the other hand, in Italian, private is the past participle of deprive.

Free software as a Commons

"Thanks to its characteristics, free software is a distributed property that is capable of evolving into a Public Good."

But, What is Free Software? And how it works? What are its inner characteristics? Just in one sentence? Why Free Software can be considered a Commons?

Its "open" and modular language, which is freely accessible and created thanks to the collaboration of many in different stages, allowing it to be perfected and modified, make the free software a "relational good" that, thanks to its accessibility, non-exclusivity and lack of competitiveness presents all the characteristics of a common resource: something which everyone can make use of, even if they have not participated directly in its creation.

Free software as an "environment for interaction". Free software *presents itself as a meeting place for scientific research, social cooperation and innovation. Thanks to its "openness", free software is capable of "evolving" as an incubator for ideas and relations, which are the abstract constituents of highly evolved technological products.*

Free software acts as an incubator. *Offering tools and resources to produce a social capital and a repertoire of information and relations* Free software *present itself as a resources available to the collectivity. This offers single enterprises the opportunity to organise themselves and work in unison.*

Free software as a sharing tool. Free software is a s*pace-environment-instrument for sharing, collaboration and commerce, because is oriented towards the production and development of other softwares*. Free software infact *facilitates the coming together of demand and supply, the interaction between producers and users and encourages the creation of products that are flexible and adaptable to the needs of multiple agents and different markets.*

Free Software favours evolution based variability. Free software *can be conceived as a "vivisystem" where the genetic pool of a software expresses its phenotype as digital lifespecies through the interaction with its socio-technical environment. The lifespecies that best adapt themselves to the environment are reproduced and spread, so they carry their genetic type to the next generation according to the rules of a sort of natural selection. Again, similarly to natural selection the mutations that occur in the software are those which reveal themselves most efficient in relation to their environment. They also decide the fitness of the artificial organisms, the software itself.*

The Value of Self-organisation

The study of emerging systems has shown that complex activities develop from single behaviours and that these are somewhat different from their simple sum.

From the behaviour of bees and ants to the free, open source software community that is responsible for developing the software, the principle of self-organisation presents itself as the dynamic element in the development of recognizable macro-behaviours, which allows for the successful achievement of complex objectives.

Similarly to insect communities and "smart mobs", Foss programmers communities are complex and adaptable systems capable to show emerging behaviours developing a peculiar movement from lower

level rules to higher and more sophisticated levels.

With time, these behaviours can increase in intelligence and respond to specific necessities that are present in thei surrounding environment.

This occurs because we are dealing with complex systems, in other words, systems with multiple agents that interact dynamically in different ways, following local rules and are able to co-evolve in symbiosis with the environment itself.

A system is emerging if produces a recognizable "macro-behaviour": in vivisystems, the most common "macro-behaviour" is **cooperation**. Indeed, symbiosis and cooperation are observed at all levels of life, from cells to complex societies.

If, within a colony of ants, single behaviours are initiated by the DNA and the objective of cooperation is to safeguard the genetic pool of the colony, in a community of software programmers, behaviours are informed by shared cultural regulations and the objective is to produce an increasingly "intelligent" software.

The cultural rules of the developers' community, which follow the method of open source, invariably go back to the strategy of cooperative altruism and hence, to the *gift logic*. This is a consolidated method within the scientific community that has, in time, been configured in a similar way to a gift economy.

The g*ift economy*, contrarily to what one may think, is an adaptive and complex behaviour with a highly rational nature. Particularly in a context characterized by a high rate of competitiveness and an abundance in resources/knowledge.

Those individuals who cooperate by following the logic of gift, compete most favourably and achieve better results than those who do not. The evolution of the GNU/Linux makes up the best example. GNU/Linux is an efficient technological system, whose added value is given by cooperation.

The mechanism of exchanging gifts, which is at the basis of free software production, facilitates the process of accumulating global knowledge. The latter, differently from many other goods, is not deteriorated by the circulation process. On the other hand, its use and consumption enhances its quality and the opportunity to create new products. This is exactly what occurs with the *Commons*. *A* **common good** is in fact a good that increases through consumption.

Cooperation, which is at the basis of free software production is a good example of the production of "digital commons" that "behave" themselves differently from the majority of other collective goods, characterised by a life span and level of consumption beyond which the resource cannot reproduce itself. Free software is just the opposite.

The mechanism of free software is a typical example of the emergence of cooperation within a complex, social supply chain whose parts autonomously concur but are coordinated to determine a result that is incommensurable to the sum of the actions by the single participants.

The emergency of cooperation

Why should cooperation be necessary? Studies on cooperation have been carried out and show that, whenever two subjects in competition (such as stakeholders with different interests) are given the opportunity to cooperate, the results of that cooperation are to the advantage of both.

Game theory has attempted to provide an explanation to this behaviour, showing that even in highly competitive environments such as in "the prisoner's dilemma", cooperation based on reciprocity is the winning strategy. This strategy, referred to as *tit for tat*, presupposes that one should begin by cooperating and then by replicating the partner's behaviour, hence maximising the results of cooperation.

Whereas in game theory, cooperation is an "economical" choice to maximise individual outcomes, in a **gift economy**, cooperation is based on **solidarity**, which is subordinate to a series of social obligations and non-economical factors, which guarantee that the community will benefit from the cooperation. The gift economy is not so much associated with the idea that things are free as it is based on a different model of exchange: the model of reciprocity. Reciprocity is the ring that joins together competitive cooperation and cooperation based on generosity.

Reciprocity that lies at the basis of gift-exchange implies a triple obligation: to give, receive and return. Giving a gift means also assuming that you will, at some point, receive a gift in return. Hence, although it is not immediate, there exists a convenience in donating a gift. One is both obliged and seeking an interest when offering a gift.

In the case of the free software, for example, the innovative contribution that each person gives may be motivated by the intellectual challenge of producing something different and useful. In other words, these is a will to create a social tie, which is geared towards advancing your local community and advertising the product to facilitate its entrance in new market fields.

However, this model of exchange, which in the case of free software leads to an accumulation of richness, is influenced by non-economical factors, such as trust in collaboration.

In the production model of the open source software, the logic of cooperation between egoists, which is typical of Game Theory, i.e. to cooperate only "if and when the other cooperates" changes to: "I will cooperate so long as the other cooperates too."

The strategy of donating gifts, until the other continues to donate gifts, implies a trust in the other person's behaviour or the certainty that you will be anle to induce a collaborative behaviour.

The precondition for such a thing to occur is that there must be a likelihood that interests will communicate and have numerous occasions to meet.... Too be continued