
Software Law

François Pellegrini
Professor, University of Bordeaux
francois.pellegrini@u-bordeaux.fr

This document can be copied and distributed freely and free of charge, provided that its content is not modified in any way, and in particular that the name of its author and its institution remain, as well as the present text.

The digital revolution

A long story, that just started

Revolutions of knowledge

- Revolution : « Abrupt economic, moral, and/or cultural change, that occurs in a society »
- Historically, the digital revolution follows two previous revolutions in the field of knowledge:
 - Revolution of writing
 - Revolution of printing

Revolution of writing

- Arising from the fixation of information on a mobile support
- Creation of the first city-states and the first great empires
 - Invention of the administration

Revolution of printing (1)

- Arising from the mechanization of copying of information on a medium
- When books were copied by hand:
 - Knowledge circulated little
 - Some rare scholars
 - From library to library, often in monasteries
 - Strong control over the content

Revolution of printing (2)

- Consequences of the mechanization of copying:
 - Low cost of access to knowledge
 - Massive dissemination of new ideas
 - Sometimes contrary to the established order
- Ferment of reforms
 - Religious reform
 - Publishers in Switzerland and the Netherlands
 - Age of Enlightenment
 - Industrial Revolution
 - Republican revolutions (circa 1750-1790)
 - The Springtime of the Peoples
 - With the massification of the distribution of books

Revolution of printing (3)

- **Pyramidization of the dissemination of information**
 - Some decide on what the multitude consumes passively
 - Big media: printing, radio, television
- **Construction of modern nation-states**
 - Standardization by language
 - Uniformisation by spreading of a “reference speech”
 - Technical standards
 - Typewriter keyboards

Industrial revolution (1)

- Its driving force and object is the machine
- Before machines, tools existed
 - Allowed one to move some matter
 - Powered by animal power
- With machines, one uses fossil energy to increase one's capacity to act on matter
 - Coal and iron → first machines
 - First machines → more coal and iron
 - Chain reaction

Industrial revolution (2)

- Two centuries later, we live in a world in which the majority of our physical actions are delegated to machines:
 - Move ourselves (cars, elevators, etc.), wash our clothes, our dishes, etc.

Digital revolution (1)

- Arising from the extraction of information from its physical medium
 - To “digitize” is to turn into digits and numbers
 - Into “heaps of 0's and 1's”
- Using tools to extract information from the physical world and code it in digital form
- A medium is always needed, but not unique

Digital revolution (2)

- Made possible thanks to two major innovations:
 - Software
 - The Internet

Digital revolution (3)

- Software is the first mechanized tool that is an extension of the human mind rather than of its body
 - Revolution in the way of producing and processing knowledge

Digital revolution (4)

- Software plays, for the digital revolution, the same role that the machine played for the industrial revolution
 - It is its driving force and object
- Same chain reaction effect:
 - Production of software → exchange of more ideas
 - Exchange of more ideas → production of more software
- Delegation of our intellectual processes to software

Digital revolution (5)

- Digital tools de-pyramidize and re-horizontize information exchanges
 - Everyone can communicate with everyone
- Disrupts the economic model of the production and diffusion of knowledge and digital works
- Creates new tools and uses
 - Majority of self-produced contents
 - We are all authors
 - Collective creation of informational goods (commons)

Digital revolution (6)

- Vast economic and social upheavals induced by the use of digital tools
- Many open issues :
 - Internet governance
 - Anonymity and digital identity
 - Status of personal data and “right to be forgotten”
 - ...

Some notions on the economy of immaterial goods

To copy is not to steal...

Economy of information goods (1)

- The economy of immaterial goods fundamentally differs from the economy of material goods
- One cannot understand the digital revolution and the profound changes it brings about, if one remains trapped in the ways of thinking of the previous millennium

Economy of information goods (2)

- Unlike material goods, immaterial goods are non-rival
 - We are not in rivalry to consume them
 - We can give them without getting poorer
 - When we “give” an idea, we copy it!
- The notion of “theft” is not relevant
 - No “owner”
 - The term “intellectual property” makes no sense
 - Various incriminations: counterfeiting, non-compliance with purpose, parasitism, etc.

Economy of information goods (3)

- The cost of copying (marginal cost) of digital goods is null
- Marginal cost: cost of copying an additional unit of some good
 - Material good: prototype → mass production
 - Immaterial good: copy at zero marginal cost
 - Even though computer and electricity are rival goods
- A digital good (software, book, music) can be distributed for free once its development has been funded

Economy of information goods (4)

- Network effects are paramount
 - The value of a product increases with the number of people using it
 - Telephone
- Example of social networks
 - Rivalry with respect to available time
 - You go to the social network where you will have the most of interactions
 - Attention economy

Economy of information goods (5)

- Very high market volatility
 - While software does not wear out...
 - Very quick obsolescence
- Strategies based on the creation of “communities”
 - Of users, of creators, etc.

The Internet

What the Internet is (1)

- A network
 - Used to transport
- A digital network
 - Used to carry information (“0's” and “1's”)
- A global network
- A public network
 - And indeed it is unique
- A network of networks

What the Internet is (2)

- “Internet is THE global public network that interconnects all networks of those who wish to participate”
 - Analogous to the road network
 - Financed by user subscriptions
- The Internet is a common good
 - Question of its status
 - Question of its governance

What the Internet is not

- One should not confuse the Internet with the services it routes
 - When we say we “go on the Internet”, in fact, most often, we use the web (or a web service)
 - Internet dates from the late 1960s - early 1970s, while the web dates back to 1992
- Internet supports many services
 - E-mail, web, videoconference, clock, etc.
 - Similar to the road network, used to implement taxi services, pizza or furniture delivery, ...

A brief history of software

And there was Software!

Informatics (1)

- Informatics is the science of efficient information processing
- Informatics is no “computer science”!
 - “Informatics is no more the science of computers than astronomy is the science of telescopes”
 - The term “computer science” is wrong

Informatics (2)

- Informatics is a science at the crossroads of:
 - Mathematics: logic, algorithmic, complexity theory, graph theory, etc.
 - Electronics: construction of computers, of networks, etc.
 - Physics: material science, magnetism, quantum physics, etc.
- Not to be reduced to “technique” or “usage”

Birth of the software (1)

- The history of computing is traced back to the first “programmable” machines
 - Mechanical looms, automata, and musical instruments
 - First uses of punch cards
- Machines not specifically dedicated to information processing
 - More “controllable” than “programmable”

Birth of the software (2)

- The first machine designed as a computer was Charles Babbage's “analytical engine”
 - Designed in 1834, built in... 2002!
 - Notions of “mill” (arithmetic unit), of “store” (memory) and of printer
 - Programs stored as punch cards
 - Ability to perform symbolic computations
 - Ada Lovelace, first program writer ever

Birth of the software (3)

- Alan Turing theorizes the notion of computer as a “universal machine” for the processing of symbolic information in 1936
 - Representation in the form of a finite state automaton
 - Program is separate from data
 - Emergence of notions of computability, complexity
- The first electromechanical computers did not have software as such
 - Programmed “by hand” through rewiring

Birth of the software (4)

- The software and the computer appeared with the idea that the program could be stored in the same memory as that used for the data it handles
 - “Von Neumann” architecture
 - Software is data (almost) like any other!
 - Enables self-modifiable programs
 - Meta-programming: viruses, etc.

Birth of software law (1)

- As long as the software was specific to a given computer, the question of its status did not arise
 - Software was provided free to users as “ancillary supplies”
 - In the same way as the user manuals

Birth of software law (2)

- Vendors encouraged their customers to modify software and share changes
- Support for the creation of “user clubs” to share these improvements
 - Mutualization and therefore reduction of software maintenance costs, offering a competitive advantage over competitors
 - The principles of free software before time!

Birth of software law (3)

- The question of the legal status of the software arose when the first IBM-compatible mainframe computers appeared in the late 1960s
 - Compatible hardware vendors used existing IBM software without paying the price
- IBM decided to charge for software and hardware separately
 - “Unbundling” policy
 - But how to legally characterize software?

What status for software? (1)

- Three paths could be followed by the legislator:
 - Patent law
 - Considered unsuitable given the object to regulate and the heaviness of the mechanisms of filing and maintenance
 - “Sui generis” right
 - Custom cut
 - Delays of implementation of international conventions
 - Author right / copyright
 - Similar creation process between software and literary works
 - Existence of the Berne Convention (1886)
 - Automatic and immediate international protection

What status for software? (2)

- Author right was chosen as the vehicle of software law
 - 1980 in the US (amendment to the Copyright Act)
 - 1985 in France (law of 3rd of July)
 - 1991 in the EU (directive 91/250/EC)
 - 1994 among WTO members (TRIPS agreement)
 - 1996 worldwide (WIPO WCT)

A few notions in informatics

Computer (1)

- “Universal Programmable Data Processing Machine”
- Machine: is driven by an external source of power
- Programmable: its operation is modifiable and configurable without having to modify the machine

Computer (2)

- Universal: can perform all proven treatments as calculable
- Information processing: manipulates symbolic values, decoupled from any physical reality
 - Most often, “0's” and “1's”
 - Corresponds to two states of matter:
 - Electric power / no electric power
 - Light / no light
 - Etc.

Computer (3)

- To fulfill its function, a computer must possess:
 - A processing unit (processor), to run the program that is provided
 - A memory, to store the program and the data on which it operates
 - Input devices, to convert information from the physical world into symbolic information
 - Output devices, to convert symbolic information into actions on the physical world

Data

- “Piece of information that makes sense to the person who produced it”
- “Elementary description of a reality”
- When more than one data needs to be processed together, they can be organized using a data structure
 - E.g.: Position = (Latitude, Longitude)
 - Aggregation of basic types defined by a grammar
 - Called “abstract type” in informatics
 - Data format + rules of processing

Algorithm

- “Set of steps to achieve a given result from provided elements”
- Used in many areas:
 - Cooking recipes
 - Chicken Tikka Masala
 - Story plots
 - The Little Red Hood
 - Mathematical methods
 - GCD calculation
- Belong to the realm of ideas
 - Belong to the common pool

Computer program

- “Expression of one or more algorithms in the form of texts written in one or more computer languages and intended to be interpreted by a computer”
- Any textual communication of an algorithm is reducible to a program
 - In a computer or human language
 - Logigrams (flowcharts) are mixed graphical / textual expressions
 - Little used because very little expressiveness

Author right

... or, more precisely, a brief overview of it

Rights of immaterial goods

- Defined in France in the “Code of Intellectual Property” (CPI)
 - Unsuitable term, because it is an oxymoron
- Hierarchical organization of the Code:
 - Code of literary and artistic property
 - Author right
 - Rights “neighbor to the rights of the author”:
“neighboring rights” / “related rights”
 - Performers' right
 - Code of industrial property
 - Patent law
 - Trademark law

Justification of author right (1)

- Encourage creation at the global scale, by guaranteeing creators a temporary monopoly on the exploitation of their creations
- At the end of this monopoly, the creations move to the public domain, and can thus benefit to all
 - A work does not “fall in the public domain”, it raises to it!

Justification of author right (2)

- The temporary monopoly granted under author right represents a compromise between authors' interest and public interest

« Le livre, comme livre, appartient à l'auteur, mais comme pensée, il appartient [...] au genre humain. [...] Si l'un des deux droits, le droit de l'écrivain et le droit de l'esprit humain, devait être sacrifié, ce serait, certes, le droit de l'écrivain, car l'intérêt public est notre préoccupation unique, et tous [...] doivent passer avant nous. »

Victor Hugo,
Opening speech of the International literary congress, 1878

Structure of author right

- Hierarchical organization of author right:
 - Patrimonial / economic / property rights
 - Attached to the work
 - Extra-patrimonial rights, also called moral rights
 - Attached to the person of the author (or his descendants)
 - Do not exist in the copyright system

Work (1)

- The work (“*œuvre*”) is a creation of form
 - The form that will be protected, not the underlying ideas and concepts
- “Ideas are of free ride!”
 - « *Les idées sont de libre parcours* » (H. Desbois)

Work (2)

- Non-exhaustive list of works deemed to receive copyright protection:
 - “Notably” (L.112-2 CPI) : books, pamphlets and other literary, artistic and scientific writings; lectures, speeches, sermons, pleadings and other similar works; dramatic or dramatico-musical works; choreographic works, circus acts and performances, pantomimes, etc...

Criteria for coverage

- Concept of “originality”, reflecting the “personality of the author”
 - Sometimes reworded as “intellectual contribution” in the case of software works
- A non-original creation of form will not be susceptible to protection:
 - E.g., a plain photograph of a painting will create no right for the photographer
 - While a blurry picture may do so!
 - E.g., “non-artistic” photographs of cooking recipes

Irrelevant criteria (1)

- “Novelty”
 - E.g., “Yet another story of musketeers?!”
 - E.g., different web browsers are independent, original works
 - While they have to comply to very restrictive standards in terms of behavior

Irrelevant criteria (2)

- “Merit”
 - Who still remembers the “official” painters from the early XIXth century?
 - Hopeful fate for many of today's “conceptual artists” ;-)
 - In the United States, rights are granted “To promote the Progress of Science and useful Arts”
 - Injunction to the lawmaker
 - Misinterpreted by some judges
 - Some works, considered as “indecent”, have been considered as unworthy to receive copyright protection

Creations escaping coverage (1)

- A creation of form that does not reflect the personality of its creator is not eligible to coverage by author rights
 - E.g., mathematical tables, tide tables, etc.
- However, graphic presentation of such a creation might be eligible if it is original
 - E.g., logos, decorations, etc.

Creations escaping coverage (2)

- Some creations that are indeed works also escape coverage since their inception
 - Part of the “initial public domain”
 - E.g. Art. 9 Romanian DDA: “The following shall not benefit from the legal protection accorded to copyright: (a) the ideas, theories, concepts, discoveries and inventions contained in a work [...]; (b) official texts of a political, legislative, administrative or judicial nature, and official translations thereof; [.../...]

Creations escaping coverage (3)

- [.../...] (c) official symbols of the State, public authorities and organizations, such as armorial bearings, seals, flags, emblems, shields, badges and medals; (d) means of payment; (e) news and press information; (f) simple facts and data.”
- It is the case for official texts of judicial nature, yet not for some technical standards!

Common pool

- The common pool contains all publicly available knowledge
 - Does not concern works, but “archetypes”, i.e. “algorithms of the real world”
 - Not to be confused with the public domain of the works of the mind
- The mathematical algorithms used in informatics belong to the common pool

Automatic coverage (1)

- Coverage is acquired since the conception of the work
 - Art. 1-1 DDA: “The author right copyright in a literary, artistic or scientific work and in any similar work of intellectual creation [...] vests in the author and embodies attributes of moral and economic character.”
 - Art. 1-2 DDA: “A work of intellectual creation shall be acknowledged and protected, independently of its disclosure to the public, simply by virtue of its creation.”

Automatic coverage (2)

- No registration necessary
- It is wise to build up in advance proofs of authorship and anteriority:
 - Bailiff's report
 - Deposit in a notary
 - Registration with specialized bodies
 - E.g., authors' associations
 - Self-addressed unopened postal mail
 - etc.

Patrimonial rights (1)

- Materialization of the right granted to the author to benefit from the uses of his/her work
 - Economic protection of the work
- Distinct from the physical possession of the work
 - One only buys the “matter” of a painting, not what it represents
 - The transfer of reproduction rights must be explicit
- Patrimonial rights are:
 - Transferable
 - Temporary

Duration of patrimonial rights

- The duration of patrimonial rights depends on the countries and the situations
 - Yet global standardization under the pressure of the entertainment industry
- Currently, within the European Union, this period is 70 years after the death of the author
 - Duration incompatible with the preservation of software heritage

Extra-patrimonial rights

- Protect both the work and the act of creation
- Three distinct rights:
 - Right to recognition of authorship
 - Ghostwriters contracts have been ruled illegal
 - Right to the integrity of the work
 - Sometimes, a specific right covers the name of the work
 - Disclosure right
- Moral rights are:
 - Inalienable
 - Perpetual

“Exceptions” to author rights

- **Defined by law**
 - They are not individual tolerances, but positive rights!
- **Exhaustive list, including:**
 - Private copy for the copy maker's sole use
 - Use in the “family circle”
 - Citation
 - Parody
 - Etc.

Work created by several authors (1)

- Three legal categories
 - Collaboration work
 - Work created by several co-authors in collaboration
 - Shared rights
 - Composite work (or “derived work”)
 - New work to which is incorporated a preexisting work, without the collaboration of the author of the latter

Work created by several authors (2)

- Collective work
 - Work in which the personal contributions of the co-authors form a whole, without it being possible, in view of the nature of the work, to ascribe a distinct right to any one of the co-authors in the whole work so created.
 - Patrimonial rights belong to the person at the origin of the work

Synthesis of legal concepts

- Legal concepts of intellectual creation

Common pool: abstract entities

- algorithms, functional and interface specifications, etc.

Creations of form

Non-original

creations:

- tide and
mathematical
tables,
implementations
of interfaces, etc.

Works of the mind

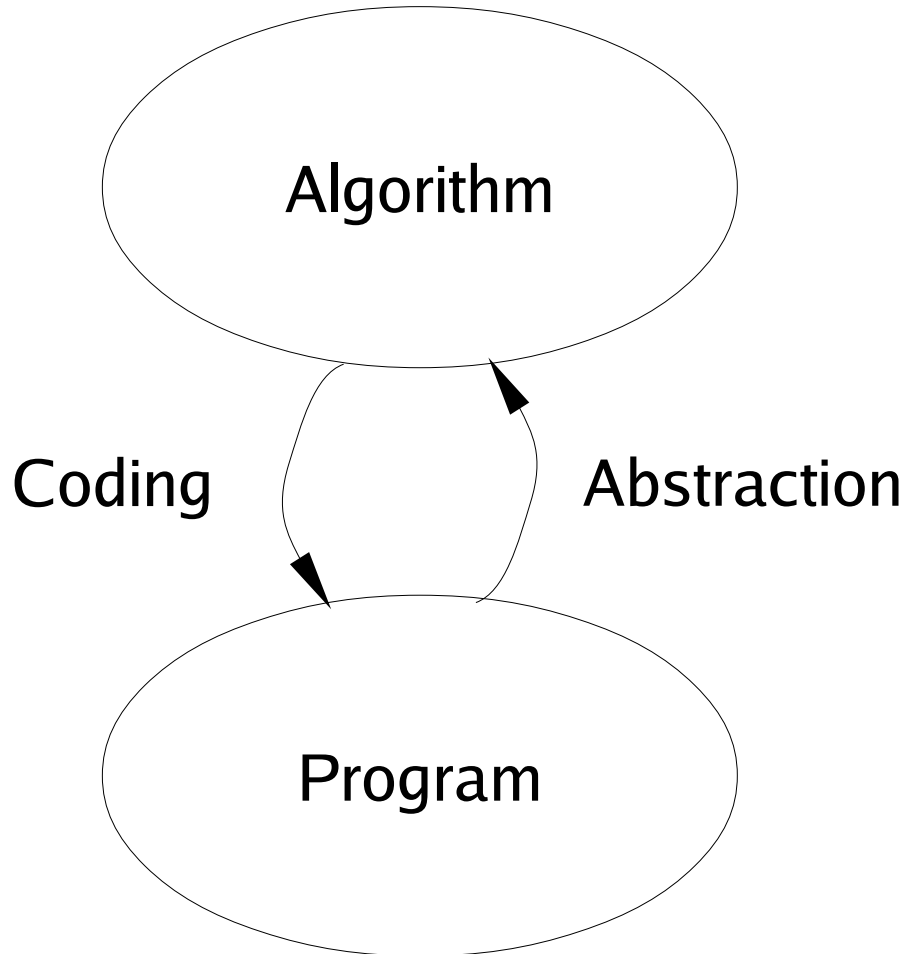
Entered/placed
into the public
domain

Covered by
patrimonial rights
(libre and private
licenses)

Author right adapted to software

That should fit in there...

Nature of software



- Algorithms are:
 - Ideas
 - Mathematics
- Programs are:
 - Works of the mind
 - Speech
 - Human ↔ human
 - Human → computer
 - Processes, whenever executed on a computer
- Similar to the process of literary creation

Adapted author right (1)

- By its integration in author right, software is considered as a work of the mind
 - French law of 3 July 1985
 - EU directive 91/250/EC (1991)
 - Luxembourg law of 24 April 1995
 - Article 10 of WTO TRIPS (1994)
 - Article 4 of WIPO WCT (1996)

Adapted author right (2)

- However, software is also a substitutable good destined to provide a service
 - Adaptation of author right (“adapted author right”)
 - Question of guarantee

Adaptation of patrimonial rights (1)

- Addition of new “exceptions”:
 - “Acts necessary to use the software”
 - “Observation of the operation of software”
 - Reminder of the spirit of the law
 - Reserve copy
 - If no other means provided by the publisher
 - De-compilation
 - Forbidden, save for “purpose of interoperability”
- Removal of the exception of personal copying
 - Need for a license for each copy of the software

Adaptation of patrimonial rights (2)

- The patrimonial rights of authors that are employees or civil servants are automatically transferred to their employer
 - It is up to the employer to decide on the life of the work:
 - Disclosure, choice of license, etc.
 - Apparition of a status of “proletarian author”
 - Already started with the notion of “collective work”
 - Only concerns software created in the professional context, during work time!

Adaptation of patrimonial rights (3)

- Authors that are not employees retain the patrimonial rights to their software works
 - Need to transfer the patrimonial rights of subcontractors
 - Problem of unpaid interns

Adaptation of extra-patrimonial rights

- Weakening of moral rights
 - Employee or civil servant authors cannot oppose the modification of their work
 - Only remains the right to recognition of authorship
 - More commonplace in the world of video gaming than in that of accounting

Scope of protection (1)

- Is covered by software-related author right anything that comes under the expression of mathematical algorithms
 - Source code and object codes that can be derived from it
 - Formal expression of what the software does
 - Expresses the personality of its authors
 - “Preparatory design material”
 - Formal expression of what the software should do
 - Does not necessarily express originality of form and therefore insusceptible of protection by plain author right

Scope of protection (2)

- Documentation benefits from a hybrid status
 - Original aspects covered by plain author right
 - Non-original aspects covered by adapted author right
- Graphic and sound elements are covered by plain author right
 - Just as there are different rights for the authors of a text and of its illustrations

The case of languages

That does not fit in there...

Language

- “System for communicating a thought”
- Written languages communicate this thought in the form of texts
- A written language is defined by:
 - A lexicon: set of words admitted in the language
 - A grammar: a set of rules for assembling and arranging the words of the language
- A language is not a text
 - It is used to write texts

Computer language (1)

- Human languages are ambiguous:
 - “*Pierre prend la boule et la lance*”
 - “*Time flies like an arrow*”
 - Etc.
- Computer languages have been designed to be unambiguous: a text has only one meaning
 - “Language theory” allows one to prove it

Computer language (2)

- A computer language is also a language of human communication
 - A programmer, reading a colleague's program, can understand the meaning of it and even detect and fix mistakes (“bugs”)
- There exist many computer languages, each having a specific way of thinking or expression
 - E.g.: C, Python, Java, Ruby, HTML, etc.
 - Analogous to various professional jargons

What status for languages? (1)

- A language is not a work
 - It allows one to write works
 - It is of a higher abstraction level
 - It is a system of thought and not a creation of form expressing a thought
 - No claim under the regime of author right
- A computer language is also a language of human communication
 - Allows for the exchange of information between humans

What status for languages? (2)

- Computer languages cannot be monopolized without harming higher rights
- A file format defines the grammar of a language
 - To grant patents on file formats would therefore be to be able to monopolize a language
 - Hence also, by extension, of elements of human languages

What status for languages? (3)

- The lexicon and grammar of a language can be described as texts
 - Yet these texts are not original because they unambiguously express the structure of language
- Definitions of words in a lexicon may be claimed under author right
 - If there is enough room for originality

Legal and economic organization of software creation

Because one can live (well) of one's art...

Economic model of software creation (1)

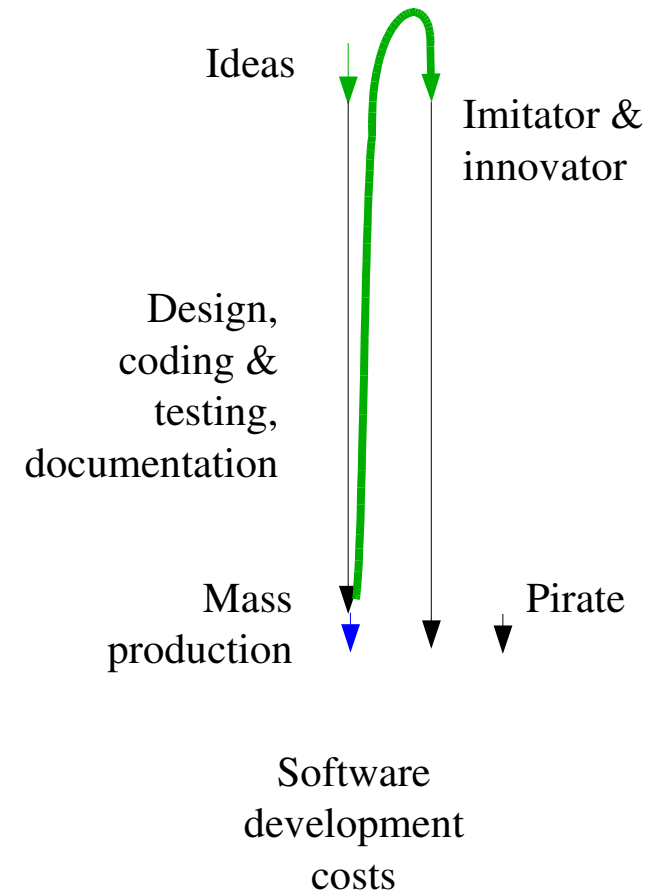
- The vision promoted by the European Commission is fully described in the recitals of Directive 91/250/EC:
 - Industrial sector of substitutable goods
 - Need to establish the rules of free and undistorted competition
 - Enforcement of authors rights

Economic model of software creation (2)

- The adaptation of the economic rights aims at adapting the modalities of economic exploitation to the specificities of this type of works
 - Individual rights granted in the service of a macroeconomic vision
 - Just as resale right is specific to plastic works

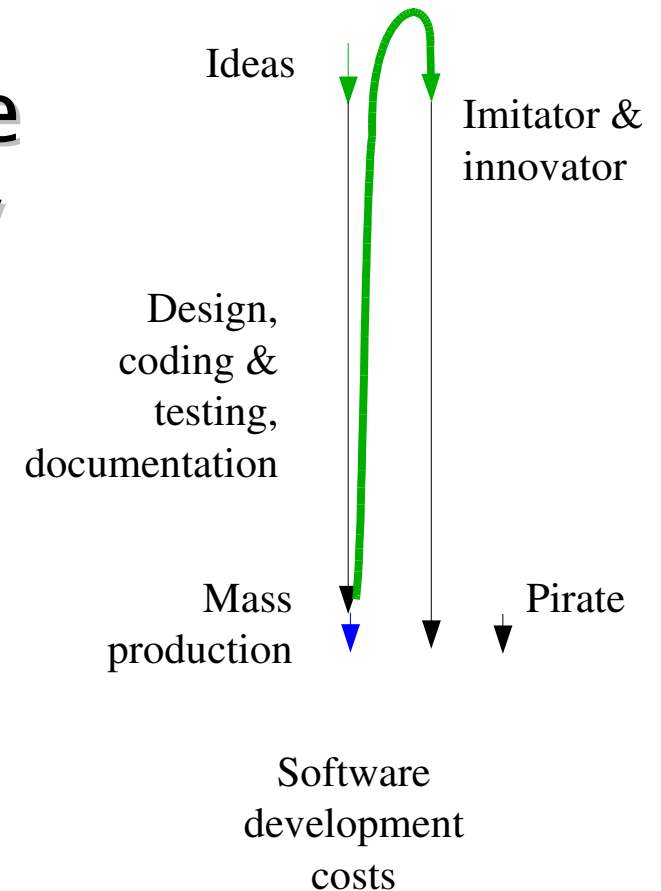
Economic model of software creation (3)

- The observation of the functionalities of an existing software allows a newcomer to create a competing software
- Meanwhile, the original innovator can/must progress
 - If it does not, it will be overtaken by the innovations of others



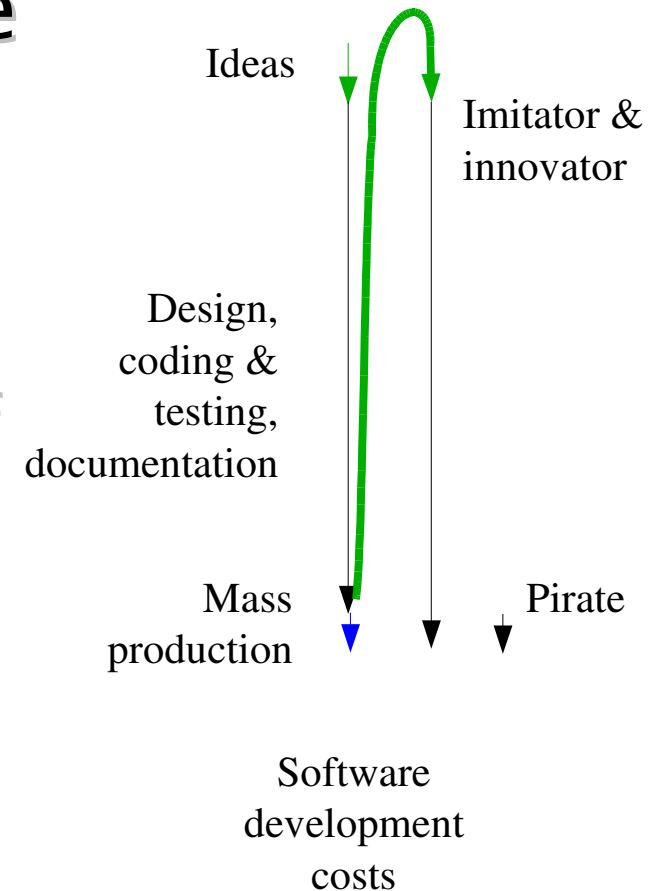
Economic model of software creation (4)

- Free and undistorted competition may exist only if the competitor pays the same entry costs on the market
 - It should not reuse the existing code, by servile copy and/or translation and/or decompilation
 - Must re-code entirely
 - However, the cost will often be lower, due to the evolution of software technology



Economic model of software creation (5)

- Need to specifically address the problem of captive markets created by data formats
 - Provisions relating to decompilation “for the purpose of interoperability”



Compilation, decompilation and interoperability

Make and break,
it's still work to do!

High- and low-level languages (1)

- Computers only understand “machine language”
 - Very rudimentary instructions
 - Specific to a given processor
 - And to a given operating system
 - Very little expressive code
 - High risk of bugs
 - Proportional to the number of lines that one writes

High- and low-level languages (2)

- Need to write programs in more expressive, so-called “higher level”, languages
 - Better understanding of the code
 - Independence from the type of processor
 - Better maintainability and portability
 - Much lower economic cost
- Automatic tools allow for the translation of texts from one language to another
 - Compilers, assemblers, etc.

Compilation and decompilation (1)

- Compiling consists in translating a program written in a high-level language into a functionally equivalent program written in a low-level language, capable of being executed by a computer
 - The “source code” is the program written in high-level language that one wants to translate
 - The “preferred form” of writing a program
 - The “object code” is the resulting program written in a low-level language
 - Defines the “executable program”

Compilation and decompilation (2)

- “Decompilation” is the inverse action of compilation, allowing one to express in a high-level language a program originally written in a low-level language
 - Much harder to implement
 - High level structural information “diluted” in object code

Automatic translation (1)

- The translation of the source code in object code is performed automatically:
 - The object code is a work entirely derived from the source code
 - No original contributions from another author
 - The semantics of the original program must be fully respected
 - The contribution is in the ideas that make an automatic translator better than its competitors
 - No rights acquired by the translator's creator on derivative works that he contributes to produce

Automatic translation (2)

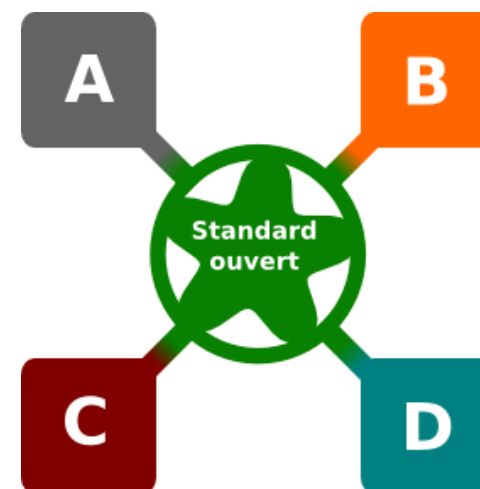
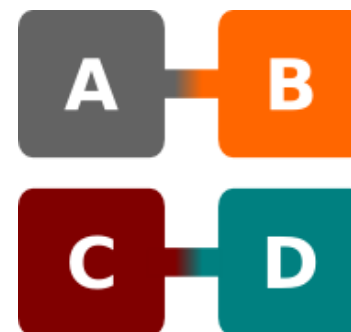
- The creator of a tool has no rights on creations made using this tool
 - Think about dictionaries, brushes, etc.
- Special case: inclusion of code fragments of the compiler in the executable program
- Question of creations produced by automated processing (“artificial intelligences”)
 - “Valladolid 2.0” !

Prohibition of decompilation

- To compile and decompile create works derived from the original work
- These copies can therefore be exploited only with the agreement of the right holder of the original works
- Decompilation is forbidden, unless explicitly authorized by the right holder

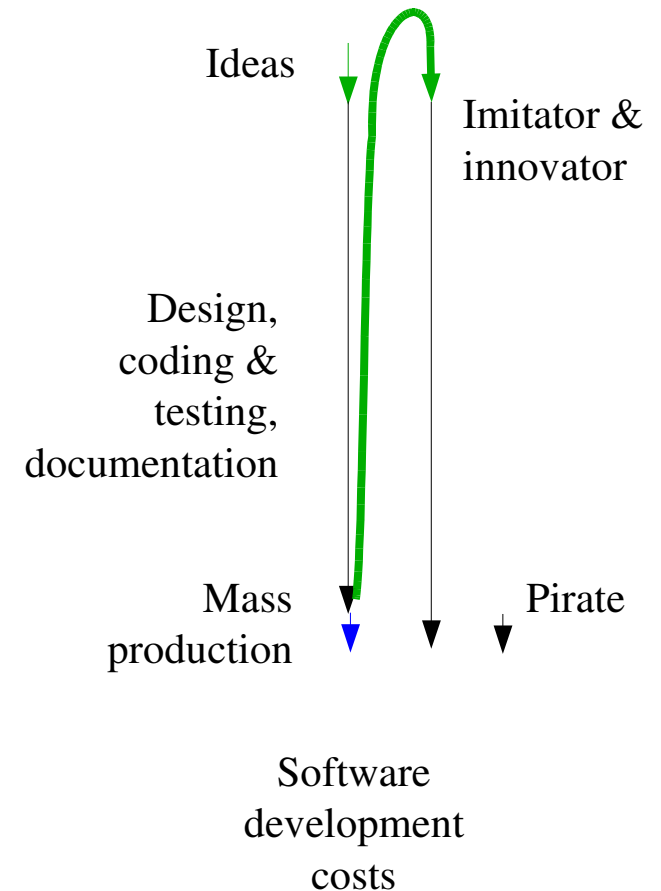
Compatibility is not interoperability!

- There is compatibility when two products can work together
 - Contractual agreement between the parties
 - The entity controlling the “compatible” format extends its monopoly
- There is interoperability when two products can work together and one knows why



Search for interoperability (1)

- To foster innovation:
 - The investments of the original innovators must be rewarded
 - New entrants should not be prevented from competing
- To create free competition:
 - Slavish imitation must be prohibited
 - Redesign must be legal
- Captive markets must be discouraged



Search for interoperability (2)

- Welcome specificity of European law
 - Established by Directive 91/250/EC
- Software decompilation is prohibited except for the purpose of interoperability
(L.122-6-1 IV° CPI)

Search for interoperability (3)

- **Strict framing by three conditions:**
 - Acts performed by a person entitled to use the software or mandated for this purpose
 - Information not already available elsewhere
 - Acts limited to the necessary parts of the software
- The information obtained can not be used for other purposes and “infringe copyright”
- Need to document the decompilation process to respond to counterfeit charges

Status of interfaces (1)

- Free competition means that new software can be interfaced with existing software
 - Even replace it by interfacing with the third-party software with which this software interacts
- In order to interface with existing software, one must comply to its interfaces
 - Described in header files, class files, etc.

Status of interfaces (2)

- To interface with another software, it is necessary to copy the interface format and its naming conventions
 - Consequently, there is no originality in the interfacing code
- An interface cannot be protected
 - An interface gives access to an extension of a language

And originality in all this?

Through some case studies...

In case of dispute...

- The existence of rights is only apparent through the evidence that can be exhibited in the eyes of the judge
- Litigation arises when two parties dispute about one or more software
- In terms of author right, when can one characterize counterfeiting?
- Do not confuse issues of originality and rights ownership!
 - Plagiarism derives from an “original” work!

Similarity of features?

- A accuses B of having infringed on his software because L_B does the same thing as L_A
 - Freedom of observation of features
 - Freedom to create a software performing the same actions
 - Right to decompile to break out of captive markets created by closed data formats
 - Including code reuse!
- Invalid remedy
 - Misuse of algorithmic patents (improperly called “software patents”)

Similarity of code? (1)

- A accuses B of having infringed on his software because the code of L_B is “very” similar to that of L_A
 - Existence of a common source?
 - Existence of a data leak?
 - Industrial espionage? Unscrupulous employee? Accidental leak?
 - Enforcement of interfaces?
 - Existence of a space of formal freedom?

Similarity of code? (2)

- This is not because the code of L_B differs from that of L_A that there is no counterfeiting!
 - The code of L_B may have been obtained by (semi-)automatic translation from that of L_A
- Role of the expert in determining the degree of fortuity of code expression similarities

Determination of originality (1)

- In any case, the originality of a software work can/should always be presumed
 - The author just “did his job”
 - The equivalence of observed behaviors is not significant
- No economic interest in denying originality to some software
 - Uncertain legal basis
 - Leave the authors without protection
 - Ruling on “merit”
 - Not relevant with respect to the acts to fight

Determination of originality (2)

- Author right disputes must be resolved through the study of the source code and how it was produced
 - Qualitative and not quantitative analysis
 - No decision on the “sweat of the brow”
- No reversal of the burden of proof
 - It is up to the claimant to prove that a code was obtained unfairly, in violation of the economic model of software creation: slavish copy, automatic translation, etc.

Licenses

Code is law
Code is poetry
Code is life !

License (1)

- The license is a tentative offer from the supplier, which defines the terms of use of a work
 - The exact legal term is: “pollicitation”
- Based on author right or copyright
 - Berne Convention of 1886
- Classically, a license restricts the rights to use a work:
 - Prevents public diffusion
 - Prevents reproduction, even partial
 - Etc.

License (2)

- The legal framework of licenses may differ across countries and legal systems
- In the United States, the “license” is a text relating to copyright, therefore at the federal level
 - Uniform interpretation, unlike “contracts”
- In Europe, their legal framework is that of consumer contracts and terms of use

Software licenses (1)

- There exist many types of software licenses:
 - “private/proprietary” license: all rights are reserved by the right holder
 - Case of most of marketed software
 - In almost all cases:
 - The customer only owns the medium, not the software it contains
 - The provider can stop maintenance at any time

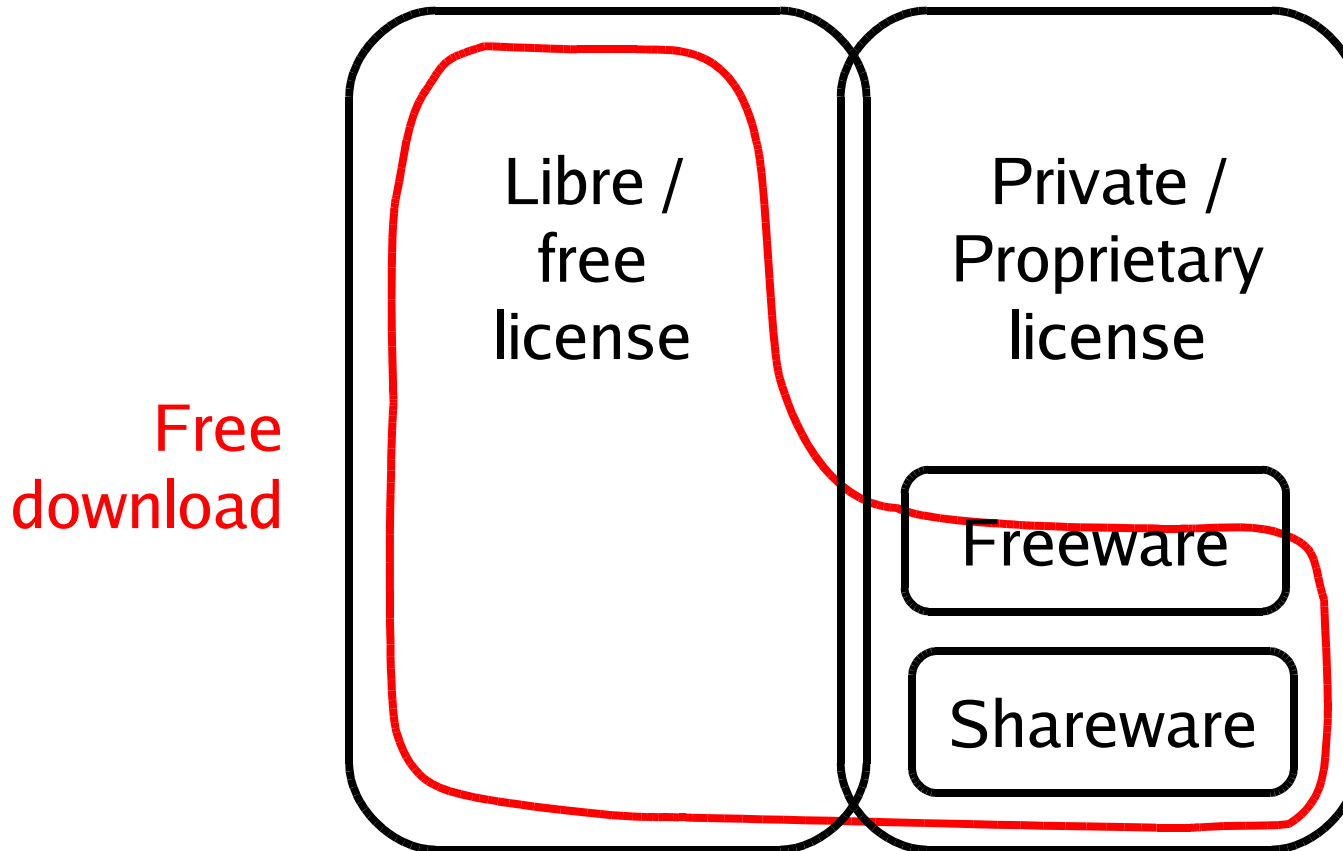
Software licenses (2)

- “Shareware” license: private software that is distributed gratuitously but for which a monetary sum is asked for after a trial period
 - Private license that differs by its distribution mode
- “Freeware” license: gratuitous private license that does not necessarily grant other rights
 - Not even the right of redistribution

Software licenses (3)

- “Free software” (as in “freedom”) license: software whose license grants many rights to its users
 - Is not equivalent to a “freeware”!
 - Not necessarily gratuitous
- “public domain” license
 - Voluntary public domain

Summary of license types



From : <http://www.gnu.org/philosophy/categories.html>

Free software licenses

Free, Free,
Set them free (ouaouuah...)

Free software

- “Free software” is a legal innovation and not a technical innovation
 - Relies on the author right mechanism applicable to software
 - Grants users some rights but also obligations
- Allows for the emergence of deconcentrated economic models adapted to the intangible economy
 - Negates the legal transaction cost
 - Technical transaction cost (marginal cost of copying) rendered null thanks to the Internet

Free software licenses (1)

- Have in common the “four freedoms”:
 - Freedom to execute the program for any use
 - Freedom to redistribute copies of the program
 - Freedom to study the operations of the program and to adapt it to one's needs
 - Need to have access to the source code
 - Freedom to redistribute the modified software
 - Capitalization of knowledge and work
 - Pooling of software developments

Free software licenses (2)

- The differences between free licenses concern the conditions of redistribution of the modified software:
 - If persistence of the obligations to provide the source code: “copyleft”
 - Collective freedom
 - If absence of this obligation: “non copyleft”
 - Individual freedom

“Free” vs. “open source” (1)

- A number of early entrepreneurs of the free software industry in Silicon Valley expressed concern that the term “free” could be perceived as “anti-business”
- Creation of:
 - The term “open source” software
 - The Open Source Initiative
 - The Open Source Definition

“Free” vs. “open source” (2)

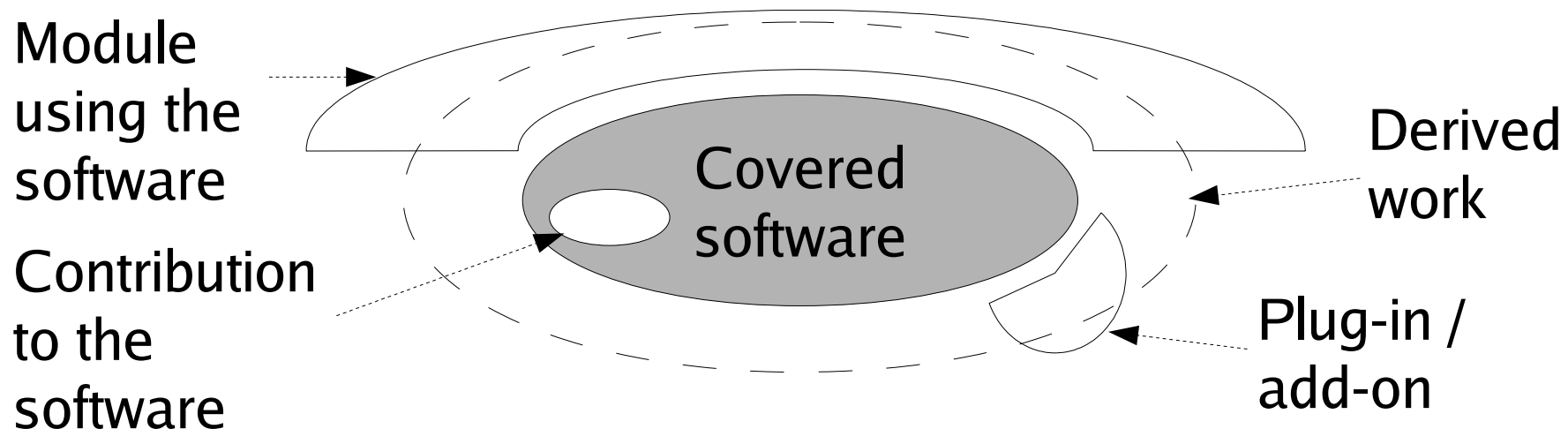
- **Open Source Definition:**
 - Free redistribution of the original code
 - Access to the source code
 - Free redistribution of derivative works
 - Integrity of the original source code (vs. patches)
 - No discrimination against people/groups
 - No discrimination against fields of endeavor
 - No additional license
 - License not restricted to a product
 - License not restricted to other products
 - Technologically neutral license

“Free” vs. “open source” (3)

- Almost full overlap between the “free” and “open source” perimeters
 - Existence of some “open source” licenses that are not considered as “free”
 - Problem now solved
- Problem of misuse of the term “open source”
 - And of “open” in general
- Preferential use of the term “free/libre”
 - “FLOSS” in some European literature

Mode of action of free licenses (1)

- The most common cases of interaction with a software module are:
 - Use: in another software or by interaction
 - Modification: work “at the heart of the software”
 - Plug-in creation: interaction through a dedicated interface (API) to extend the services provided



Mode of action of free licenses (2)

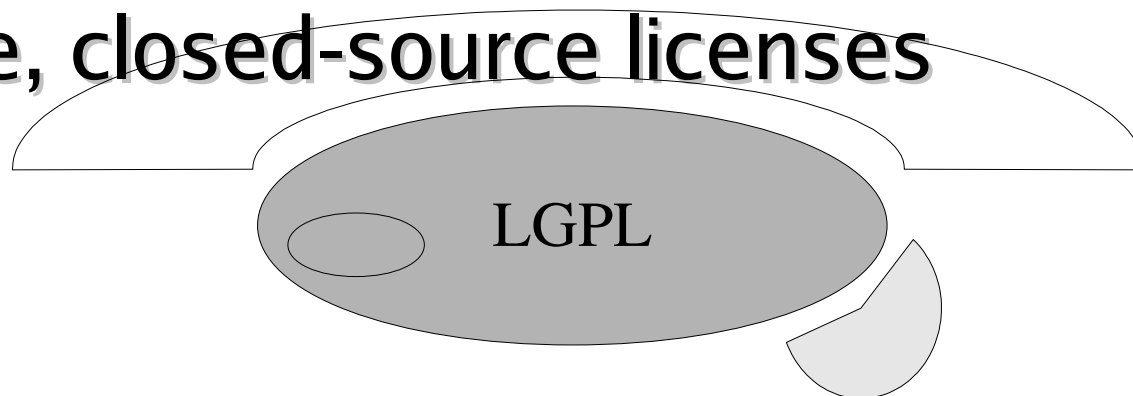
- Free licenses ensure that “their terms” apply to any derivative work
 - Used to regulate the use that is made of the module
 - Allows to regulate the choice of the license of the modules related to the covered module
- Do not always insist that the license itself must cover the constituent elements of the derivative work!
 - Depends on the diffusivity or not of the license

Main types of free licenses

- Three main types of free software licenses:
 - “Persistent” licenses
 - “Evanescent” licenses
 - “Diffusive” licenses

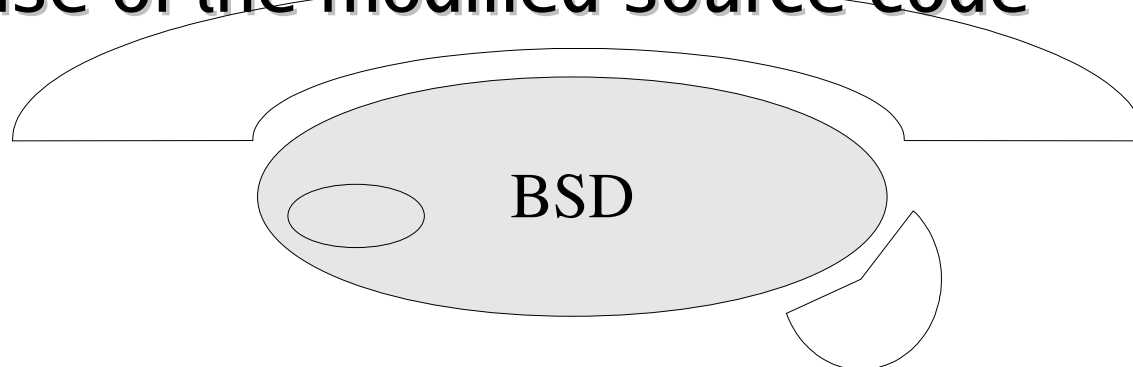
“Persistent” licenses

- Also called: “perennial”, “weak copyleft”
- Examples : LGPL, CeCILL-C
- The source code of modified versions must be redistributed when they are distributed (permanence)
- However, it can be combined with software subject to other kinds of licenses, including private, closed-source licenses



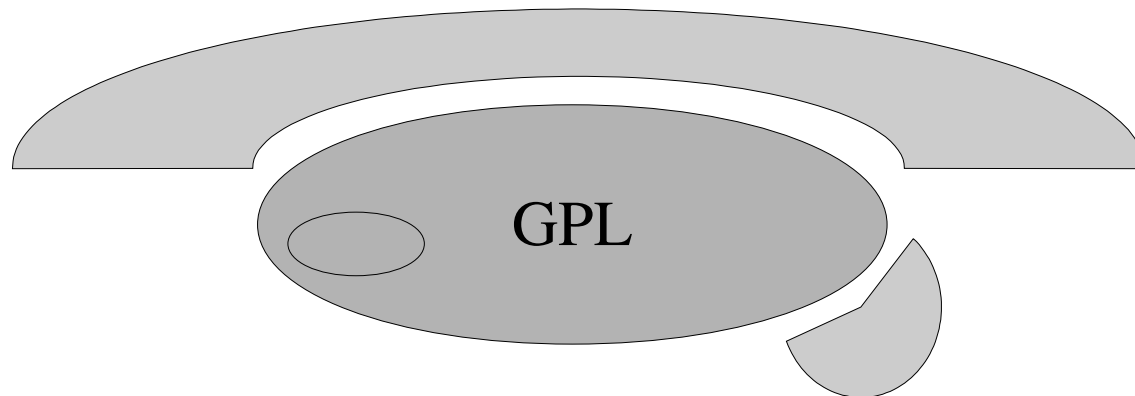
“Evanescent” licenses

- Also called: “permissive”, “non copylefted”
- Examples : BSD, CeCILL-B
- The modified software can be distributed on binary form under any type of licenses
 - A software can be “closed-up” and redistributed without its source code
 - The CeCILL-B license even allows one to change the license of the modified source code



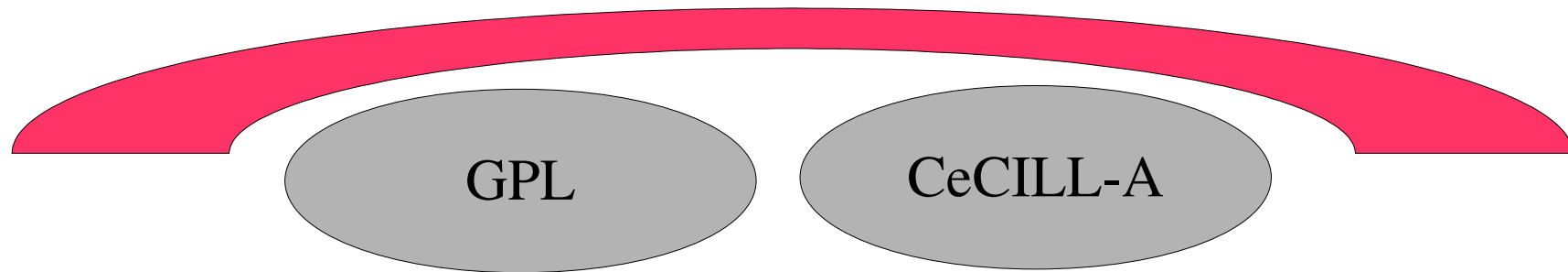
“Diffusive” licenses (1)

- Also called: “strong copyleft” or sometimes “contaminant” (biased term, to avoid!)
- Examples : GPL, CeCILL(-A)
- The source code of modified versions must be available to anyone receiving the object code
- Strongly connected third-party software have to be diffused according to the same terms



“Diffusive” licenses (2)

- Diffusive licenses can induce conflicts when one tries to link two modules covered by two different diffusive licenses



- Some “minor” licenses have clauses to explicitly solve these conflicts
 - The CeCILL-A gives way explicitly to the GPL

Multi-licensing

- The right holder of a software may choose to release it with the license of her/his choice
- It can even release the same source code, through several different channels, with different licenses
 - Multi-licensing policy (e.g., “dual licensing”)
 - Be careful with the tracking of external contributions in successive versions!

The license as a work (1)

- Licenses are texts which, albeit of a technical nature, can claim the status of a work
 - Like scientific papers
- The trustee of the license is theoretically the only one able to offer later versions of the license
 - Protection of the name
 - Re-use of terms by third parties
 - Example of the BSD-2 (“two-clause”)

The license as a work (2)

- Traditionally, the decision to republish a work under a new license rests with the right holder
- Free licenses allow to delegate “updates” to recipients of software
 - Explicit “mutation” clauses within the licenses themselves
 - Indication of ability to transition to “any higher version” or “any other version” put in place by the right holder of the software

Triggering of licenses (1)

- The first free licenses were created at a time when the uses of networks were poorly developed
 - Centralized operations of software by their users (natural or legal person)
- These licenses are triggered by receiving a copy of the software
 - Gives access rights to the source code and imposes the rules applicable to redistribution
 - GPL, LGPL, etc.

Triggering of licenses (2)

- The development of peer-to-peer file sharing systems means that people can be redistributed software without having decided to accept the license
- These licenses had to be amended to accommodate involuntary receptions, as is the case with peer-to-peer networks.
 - Difference between “*conveying*” and “(*ancillary*) *propagating*” (art. 9 GPL v3)

Triggering of licenses (3)

- The emergence of software-as-a-service (SaaS) software uses means that many people use free software without being able to benefit from the terms of their licenses
 - Remote use does not give access to the software
- New forms of licenses have been created to (partially) address this problem: licenses triggered by the use
 - AGPL v1.0 from Affero, taken by FSF as of v3.0

Triggering of licenses (4)

- However, the triggering by the use does not really solve the problems posed by the transition to a “SaaS” environment
 - Receiving a copy of the software and modifying that copy does not require the SaaS provider to modify its own copy
 - Software used in SaaS that does not allow the export of data will leave the user in the real impossibility of changing supplier

Use cases of free software licenses

New economic paradigm

- To maximize one's economic function, one must change the paradigm:
 - Mutualize fixed costs between players having the same interests
 - Focus on one's added value
- “Coopetition” model
 - Improves individual resilience by pooling risk taking
 - Building communities to increase the value of the ecosystem

Free software economy (1)

- Free software can be “commercial” software
 - Any software can be freely copied as soon as it has been funded
- Monetization of software supply
 - Obtainment of software from its initial creators or from other entities
 - Negotiation of maintenance and/or service contracts

Free software economy (2)

- Economic models based on:
 - Pooling:
 - Through offer: consortia (infrastructure software)
 - Through demand: coalitions of users (specialized software)
 - Service: maintenance, improvement
 - Yet service activities produce little new code
 - Know-how: freedom of choice
 - Free licenses make rent-seeking models impossible
 - Yet software entry costs can be important
 - The existence of a living ecosystem is of paramount importance
 - Decentralization: close to the end user

Use case: free download

- Software whose market is very big, made of non-competing entities or for which the software is not part of their core business
 - Case of libraries and infrastructure software
- Ability to create a community of users and contributors
 - The freedom of use increase their size and value
- Pooling of maintenance and of further developments
- Diffusion under persistent or diffusive free licenses

Use case: closed consortium

- Industry software whose market is narrow
- Pooling of development effort among consortium members
 - Right ownership according to the respective contributions
 - Complete freedom to involve new members to future upgrades of the common work
 - No desire to give gratuitously to competitors what cost to produce
 - “Free” does not mean “gratuitous”!
 - “Free” does not mean “free download”!

Use case: privileged partnership (1)

- Industrial software or library potentially usable by a larger community
- Selection of a privileged partner that provides a return over experience

Use case: privileged partnership (2)

- Release under two types of free licences (or a private one):
 - Software provided to the partner under an evanescent free license so that she/he can embed it into products whose characteristics are hidden to competitors
 - Free download under a diffusive license to foster contributions from the community and creation of analogous software whose source code is accessible to all

Why develop under free licenses

- Ideal tools for the preservation of intellectual heritage
- Zero cost
 - The addition of the license terms in each source file is sufficient to benefit from the terms of the license
 - Yet think about how to get proof of anteriority
- Pooling of the development effort
- Better durability for customers
 - Becomes a commercial argument

Good development practices

Prevention is better than cure...

Legal-technical issues

- Legal analysis must be concomitant with technical analysis
 - The choice of software components and their licenses determines the cost and the economic models of the realized software
 - Better to deal with it upstream than downstream
- Compatibility between licenses must be studied in detail
 - Not necessarily simple
 - Dedicated software can (partially) automate this process

Practical implementation

- Use version managers to track all contributions
 - Ensures (semi-) automatically the respect of the right to the name
 - Lets one know what to re-code in case of legal uncertainty
- Use two repositories:
 - A private repository for people whose rights are known (and controlled)
 - A potential public repository to elicit contributions from the community

Right of database producers

Database

- Defined by EU Directive 96/9/EC
- Refers to “literary, artistic, musical or other collections of works or collections of other material such as texts, sound, images, numbers, facts, and data [...], collections of independent works, data or other materials which are systematically or methodically arranged and can be individually accessed”

Database rights

- Two distinct rights:
 - Right relating to the arrangement of matters
 - Database producers (DBP) right

Right relating to the arrangement of matters (1)

- “[Benefit from the protection provided by this Code without prejudice to the rights of the author of the original works] the authors of anthologies or collections of works or other data, such as databases, which, by the choice or arrangement of matters, constitute intellectual creations” (L.112-3 CPI)

Right relating to the arrangement of matters (2)

- Difficulty of application *stricto sensu* to the structures of databases in their generality
 - Structure belongs to the field of ideas
- Clearly inadequate writing

DBP right (1)

- *Sui generis* right specific to the European Union
- Concerns producers:
 - “The producer of a database, understood as the person who takes the incentive and the financial risk of the corresponding investments, benefits from a protection of the contents of the base, when the constitution, the verification or the presentation the latter evidences a substantial investment in terms of finance, hardware or personnel [.../...]

DBP right (2)

This protection is independent from, and applies without prejudice of, rights resulting from author right or any other right on the data base and of its constitutive elements”

Prerogatives of DBPs (1)

- The database producer has the right to forbid:
 - The extraction, by permanent or temporary transfer to another medium of the whole or of a qualitatively or quantitatively substantial part of the contents of a database, by any means and under any form
 - The re-use, by making available to the public, the whole or of a qualitatively or quantitatively substantial part of the contents of a database, under any form

Prerogatives of DBPs (2)

- The database producer has also the right to forbid:
 - The repeated and systematic extraction of qualitatively or quantitatively non substantial parts of the contents of the database, when these operations clearly exceed the normal conditions of use of the database
- Aims at the targeted siphoning of specific portions of data bases
 - E.g., a city telephone directory

“Software” patents

A matter to laugh at,
If it was not a matter to cry about ...

Scope of adapted author right

- Guarantees that a software can not be used outside the terms of its license
- Does not prevent one from writing a new program:
 - Providing similar features
 - Compatible in terms of input/output formats
 - Interoperable with the original program

Rights covering software in Europe

- Computer programs are protected by author right
 - European Directive 91/250/EC
- Software is explicitly excluded from patentable subject matter by article 52§2c of the Munich Convention of 1973, like games and maths

Patents and software (1)

- However, a patentable physical process must not be excluded from patentability because it embeds software
 - Taking into account the digitization of the control of industrial processes
- Exclusion of patentability limited to software “as such”
- Allows you to claim “computer-controlled” inventions
 - But innovation does not lie in the software

Patents and software (2)

- With the move of economy towards software, patent offices have seen large parts of the economy escape their control
- Temptation to capture for their benefit part of this manna
 - Extension of the patent system to software
 - Contrary to the will of the lawmaker
- Need to bend the law, by means of two major intellectual deceptions:
 - Abusive extension of the scope of patentability
 - Confusion between ends and means

Criteria for patentability

- To be patentable, an innovation must:
 - Belong to a patentable domain
 - “Technical” domain, involving the control of “forces of nature”
 - Medical practices were excluded from patentability
 - Be new
 - No prior disclosure
 - Evidence an inventive step
 - “Not obvious to a person skilled in the art”
 - Supposed to prevent trivial patents
 - Be susceptible of “industrial application”
 - Supposed to limit the scope of patentability

Abusive extension (1)

- Arbitrary change of examination rules:
 - Old rules: “novelty” should only concern “technical” claims
 - New rules: “novelty” considered “as a whole”, that is, even if it concerns only “non-technical” features
- Presumption of existence of software “not as such” that can be patented
 - Allows to claim a new software (not “technical”) running on a computer (“technical” but not new)

Abusive extension (2)

- Newspeak term of “computer-implemented invention”
 - What a computer implements is an algorithm!
 - Using software that expresses this algorithm as a creation of form
 - Pretends that an algorithm can be an “invention” in the sense of patentability
 - While mathematics is not patentable “as such”

Confusion means / ends (1)

- Misrepresentation that software has a “technical effect”
 - That is, an action in the physical world
- A software handles symbolic values
 - Decorrelated from all physical reality
 - Executable “by hand” or in a simulator without producing such “technical effects”
- It is the computer peripherals that interact with the physical world

Confusion means / ends (2)

- The improvement of a software is not a “technical solution” to a “technical problem”
 - The software does not change the nature of the interaction between the computer, its peripherals and their environment
- Otherwise, an F1 driver would be patentable because he improves racing time (“technical problem”) with a car (“technical means”)

Confusion means / ends (3)

- Arbitrary change of examination rules:
 - Old rules: to propose a “technical solution” to a “technical problem”
 - New rules: simple use of “technical means” to solve any type of problem, including “non-technical”
- Allows one to claim “business methods”, that is, “real-world algorithms”:
 - Online commerce methods
 - Calculation of pension benefits using a computer

Scope of algorithmic patents (1)

- Algorithmic patents do not directly relate to programs
- They aim at monopolizing the underlying concepts, such as:
 - What a program does:
 - What concrete problem this program solves, that is, which business method it implements
 - How it does it, and more particularly:
 - What input data he accepts
 - What output data it produces
 - How it interacts with other programs

Scope of algorithmic patents (2)

- These patents relate to:
 - Business methods (“real-world algorithms”)
 - Steps taken by a user buying goods on the Internet (Amazon's “1-Click” patent), ...
 - File formats
 - Documents, spreadsheets, image (GIF), son (MP3), ...
 - Communication protocols between programs
 - Algorithms
 - Encryption, ...
- These concepts are logically inseparable

Examples of invalid patents granted by EPO

L'Urgence du E-Commerce Européen

http://webshop.ffii.org/

1 **Votre boutique web est BREVETÉE!**

4 Musiques **Films** **Livres**

2 **NOUVEAUTÉ: COMMANDEZ PAR TÉLÉPHONE PORTABLE!**

15 Obtenez de l'aide directement depuis nos bases de données de support internes! **?**

17  Les coccinelles sont des insectes très utiles. Elles se débarrassent des parasites. Cependant les services contentieux des brevets logiciels sont en général bien trop gros pour elles.

7 Voir dans le navigateur

8 Exclusif: téléchargez immédiatement vos achats!

6 Achetez la musique originale (mp3)

6 Achetez le film

7 Vous aimez ce résultat de recherche? Vous aimerez sans doute ceux-ci:

1. La coccinelle à Monté-Carlo
2. Coller opter
3. Coque de scie
4. Brevets logiciels et autres parasites
5. Tu me cherches des poux

19 <Entrez votre réduction>

3 Ajoutez au panier

10 Envoyez comme cadeau

11 Demandez un prêt

9 Paiement par carte de crédit

12 

16  Aperçu de quelques chapitres: cliquez dans la télé ci-dessus!

5 Rendez-vous dans l'un de nos magasins et composez/gravez votre propre DVD à la carte!

20 

13 Oui, je veux recevoir vos offres!

14

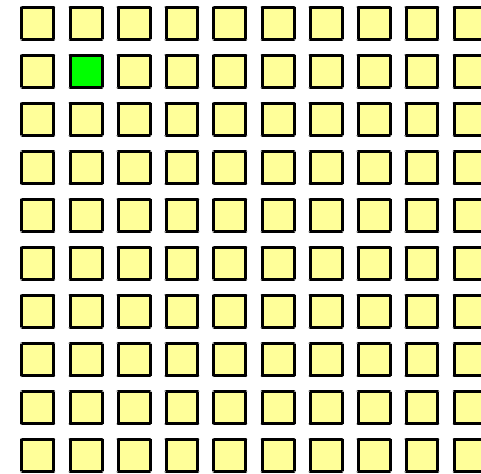
Si nous n'avons pas votre commande en stock, elle sera immédiatement transmise à l'un de nos vendeurs affiliés!

Software patent economy (1)

- Dreams of fortune

- My patented “invention”

- Obvious programming techniques

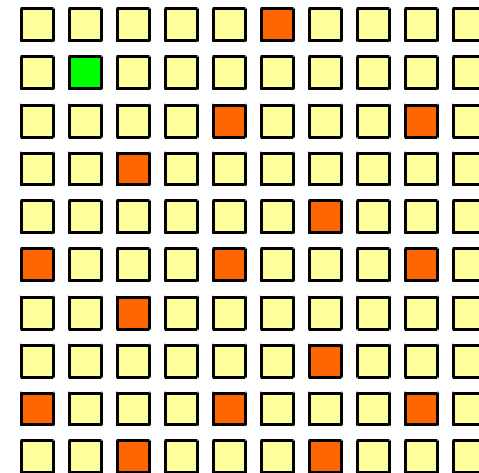


$$\text{Profit} = \text{Sales} - \text{Develop} + \underbrace{\text{Rent (■)} - \text{Cost (■)}}_{> 0}$$

Software patent economy (2)

- Software patent battles

- My patented “invention”
- Non-patented techniques
- Risk of infringement lawsuit



$$\text{Profit} = \text{Sales} - \text{Develop} + \underbrace{\text{Rent (■)} - \text{Cost (■)} - \text{Cost (■)}}_{< 0}$$

Software patent economy (3)

$$\text{Profit} = \text{Sales} - \text{Develop} + \underbrace{\text{Rent (■)} - \text{Cost (■)} - \text{Cost (■)}}_{\text{On average } < 0}$$

Small publisher:
Must pay

$$\underbrace{\text{Rent (■)} - \text{Cost (■)}}_{< 0} - \underbrace{\text{Cost (■)}}_{< 0}$$

Big publisher:
Neutralizes competitor's patents

$$\underbrace{\text{Rent (■)} - \text{Cost (■)}}_{< 0} - \underbrace{\text{Cost (■)}}_{\approx 0}$$

Patent fund:
Lives off the system

$$\underbrace{\text{Rent (■)} - \text{Cost (■)}}_{\text{In principle } > 0} - \underbrace{\text{Cost (■)}}_{= 0}$$

Who survives?

- **Big companies**
 - Cross-licensing of patent portfolios
 - Pretend to live in a world without patents
 - Oligopoly position
- **Patent funds (“trolls”)**
- **Small companies willing to get bought**
 - Their patents increase their alleged value
 - But systemic risk ignored!
- **Lawyers, patent experts, patent offices**
 - Waste between 10 and 40% of the resources initially devoted to innovation

Who withers?

- Small software publishers, generally poorly prepared for legal battles, because of:
 - Lack of funds (average lawsuit cost is 2 M\$ per side)
 - Lack of patent portfolios to share
- Companies that use « in-house » software to solve their business problems, even if:
 - They do not belong to the software industry
 - They do not sell their software
- Free software developers
 - Source code is available to the attacking party

Threats to innovation and interoperability

- If the holder of a software patent refuses to grant licenses:
 - It is illegal to write programs that read or produce data structured according to some patented format
 - Users can no longer switch to other products to process their existing data
 - Market entry of new players is reduced
 - Choice of products and providers is reduced

Assessment of algorithmic patents

- Instrument designed for the industry of material goods, abusively extended to intellectual methods
 - Monopolizes algorithms (= mathematics!)
- Illegal in Europe but granted by patent offices, that live off patent granting and renewal fees
- Economic nonsense
 - Anti-competitive instrument
 - Expensive to obtain and maintain [Bessen & Hunt]
 - Threatens return on investment of software authors
 - Promotes off-shoring