Computer Ethics

Digital intellectual property

Viola Schiaffonati

November 4th 2020

Outline (Johnson 2009)

- The **complexities** of digital property
- **Protecting** property rights
- **Free** and **open source** software
- The philosophical basis of **property**
- New **rules**

Carol and the software

- *Carol works as a **computer consultant** for a large consulting company*
- *When she goes on vacation in Southeast Asia Carol finds an **office suite package** that looks **identical** to a package **made by a well-known American company** but cost $50 (instead of $1500)*
- *She has heard that countries like the one she is in **do not honor U.S. copyright**, but the deal is just too good to resist and she **buys the software***
- *As she prepared for the airplane trip home, **she's not sure** what will happen if custom officials notice it as she reenters in the U.S.*

Carol and the software

- Has Carol done anything wrong?
- Would it be unfair if U.S. custom officials stopped Carol and confiscated the software package?

Bingo and Pete

- ***Bingo Software Systems*** *is a small company that obtained venture capital (2 millions $) and spent* ***3 years*** *to develop a* ***new file organizing system***
- *When completed, the new system is* ***successfully marketed*** *for* ***1 year*** *and Bingo recovers about* ***50% of the investment***
- *After the first year a competing company,* ***Pete's Software****, starts distributing a* ***file organizing system*** *with many of the same functions of Bingo's one that can be* ***freely downloaded*** *using a license*
- ***Pete's programmers studied Bingo's system****, adopted a similar approach, then produced a* ***new*** *piece of* ***software*** *which is* ***more efficient***
- *According to its lawyer,* ***Bingo would be unlikely to prevail in*** *a copyright or a "look and feel"* ***lawsuit*** *against Pete's software*

More on Bingo and Pete

- ***Pete’s Software*** *has plenty of* ***business****, whereas* ***Bingo*** *is unable to recover the full costs of developing its original system and files for* ***bankruptcy***
- *Customers appear to be downloading Pete’s software and then making multiple copies for internal use*
- *Some of these companies hire Pete’s Software to help them*
- *Pete’s Software, pleased by its success, begins another project in which they target another market segment currently served by proprietary software; they plan to again develop a Free Software alternative*

Bingo and Pete

- Is it situation unfair?
- Has Pete’s software wronged Bingo Software Systems?

The complexities of digital property

- **Software** as an **intellectual property**
  - Its digital composition makes software **difficult to protect** using **conventional intellectual property** regimens
- **Reproducibility**
  - Software can be **stolen** merely by **copying**
  - The owner does not lose access to the software and she may not even notice that a copy has been made
- Software has posed the more difficult **challenge** to **conventional notions** of **property**

Definitions

- Any piece of software can be described in three different ways
  - **Algorithm** is a step-by-step method for solving a particular problem
    - More abstract than a particular program written for a particular computer in a particular programming language
    - Same algorithm can be implemented in different programs
  - **Source code** refers to the program expressed in a programming language (before compilation version of the program)
  - **Object code** refers to the binary expression of that program in the machine language of a particular machine (after compilation version of the program)

Protecting property rights in software

- Three forms of **legal protection** now widely used to own and control access to software
  - **Copyright, trade secrecy**, **patent**
- However, **software never fit neatly** into these legal forms
  - Plethora of court cases and a good deal of **uncertainty** about what can and cannot be protected

Copyright

- When a software developer creates an original piece of software, he can use copyright law to obtain a form of **ownership excluding others from directly copying** the software **without permission**
- At the heart of copyright law
  - An **idea cannot be copyrighted** (algorithms)
  - The **expression of an idea can be** (programs)

Copyright: issues

- Complex issues of interpretation often arise
  - This distinction doesn’t capture **functionality** or **behavior** of software
  - Competitors can read a piece of software, comprehend its useful behavior and develops **new software behaving in the same** way but produced by entirely **different source** and **object code**

Trade Secrecy

- **Right** to keep certain kinds of **information secret**
- What is claimed as a trade secret must
  - Have **novelty**
  - Represent an **economic investment** to the claimant
  - Have involved some **effort** in **development**
  - The company must show that it made some effort to keep the **information a secret**
- **Software** can meet these requirements
- Software developers employ a **variety of tools** to **protect their secrets**
  - Nondisclosure clauses and licensing agreements
  - Limiting what is available to the user (no access to the source program) or building into the program identifying codes

Patent protection

- It offers the strongest form of protection because a patent gives the inventor a **monopoly** on the **use** of the **invention**
  - *Referring back to scenario 2, patent protection could give Bingo Software the power to prevent Pete's from marketing its system if important aspects of Bingo's system were deemed patentable*

Patent protection

- A patent claim must satisfy a **two-step test** before a patent can be granted
  - Fall within the **category of permissible subject matter**
  - It must have a) **utility**, b)**novelty** and c) be **nonobvious**
- But what subject matter is transformed by software?
- What does exactly one own when one has a patent on a piece of software?

Free and open source software

- Two-track system for control and distribution of software
  - **Proprietary systems (PS)** protected by copyright, trade secrecy, and patent law
  - Software that is produced and distributed under one of the categories of **Free and Open Source Software (FOSS)**
- FOSS vision of transparent software that **users** can **modify** to fix their needs and that is **widely available**
- FOSS programmers do nothing illegal but they make their software available to the public, often **for free** and under a license allowing users **access to the source code**

Free and open source software

- Three approaches to digital ‘sharing’
  - **Free software** (FS)
  - **Open Source Software** (OSS)
  - **Creative Commons** (CC)

Free software (FS)

- Users have the freedom to run, copy, distribute, study, change, and improve the software
- When FS is licensed, this requires that if the **code** is **incorporated into another program**, the **new program** must also **be FS**
- This has been called ‘**copyleft**’ by Richard Stallman

Four freedoms

- FS is committed to 4 freedoms for software users
  - The freedom **to run** the program, for any purpose
  - The freedom **to study** how the **program** works, and **adapt** it to your needs (access to the source code is a precondition for this)
  - The freedom to **redistribute** your copies
  - The freedom to **improve** the program, and **release** your improvement to the public, so that the whole community benefits (access to the source code is a precondition for this)

FOSS and PS

- A common misconception is that FOSS is never distributed with a cost
  - Red Hat Unix is a counterexample
- It is the **rights** that come with the software that distinguish FOSS from PS, particularly the right **to view** and **modify** the source code
  - Software developers can still make money selling and customizing FOSS
  - In the last few years also big corporations (e.g. SUN and IBM) have begun to develop and distribute FOSS
- However, FOSS represents a **threat to PS**, and some PS developers have long argued against FOSS, claiming that is unreliable and ‘communistic’

The philosophical basis of property

- Software as we are using the term here didn’t exist before computers
- Much of the struggle about what software is takes place around **social conceptions of property**
- We can distinguish two broad theories of property that are often hidden beneath the surface of debates about ownership of software
  - **Utilitarian theory**
    - The reasoning behind patent and copyright law is utilitarian: fostering creativity and innovations and encouraging disclosure
  - **Natural rights theory**
    - Proprietary rights in software are often defended as a matter of natural right

Natural rights arguments

- Because individuals own themselves, they own their labor
- **John Locke’s theory of property** (XVIII cent.)
  - Individuals have a right – a natural right – to what they produce with their labor
  - **One’s labor** is an **extension of one’s self**
  - To seize the products of someone’s labor is to render the person a slave

Natural rights and software

- **Software developers** could argue that the software they create is rightfully theirs because they produced it with their labor (both intellectual and physical)
  - *Scenario Bingo and Pete: it seemed unfair for Bingo to invest its resources and labor in software only to have it used by Pete's*
  - *Pete's Software used the work of Bingo and was able to make money from the work, yet they paid Bingo nothing*

Critique of the natural rights argument

- A first flaw in the labor theory concerns the '**naturalness**' of **the connection** between **labor** and **ownership rights**
- A second flaw relates to **software being nontangible** (intellectual) and therefore **making confiscation impossible**
  - If I labor in creating a song and someone hears the song (even memorizes it), I don't lose the song

FOSS versus PS

- Difference between PS and FOOS in terms of the **best system** for the **production** and **distribution of software**
  - 'Best system' as a matter of which produces the **best consequences**
  - Which system will create the most robust environment for software development?
  - Which system will produce the best – most useful –software?
  - Which system will lead to the most widely accessible software?
- This framework puts the focus on deciding ownership issues in terms of **effects** on **continued creativity** and **development** in the field of software
- It suggests that the courts will have to continue to draw a **delicate line** between **what should be ownable** and **what should not be ownable**

Is it wrong to copy PS?

- Question not from a legal point of view, but from a **moral** one
- Making copies of PS is not uncommon
  - It would seem that many individuals do not think it is wrong to copy PS (individuals who would not break other laws will make illegal copies of software)
- In arguing that its is wrong to copy PS without permission it is argued that it is **immoral to do something** illegal
- Does this also hold when the laws (e.g. those protecting software ownership) are **bad** laws?
- This implies that it is **permissible to break laws** whenever they are **bad**
  - A rich philosophical literature addresses why citizens have an obligation to obey the law and when citizens are justified in breaking the law

In defense of software copying

- You have to show that
  1. The system of property rights for software is not just a bad system, but an **unjust system**
  2. Adhering to those laws compels you to perform **immoral acts** or support **unjust institutions**
- If you can make the case for (1) (which is not easy), then (2) will become more plausible

Exceptions?

- Several authors have made this sort of argument although their conclusions apply only to **copying in restricted circumstances** (Stallmann 1995 and Nissenbaum 1995)
  - A person having a great deal of trouble trying with a computer and a close friend has software that will solve the problems
  - Not helping your friend seems wrong; however these authors don’t seem to recognize the harm done to copyright or patent holder when a copy is made

An interesting analogy?

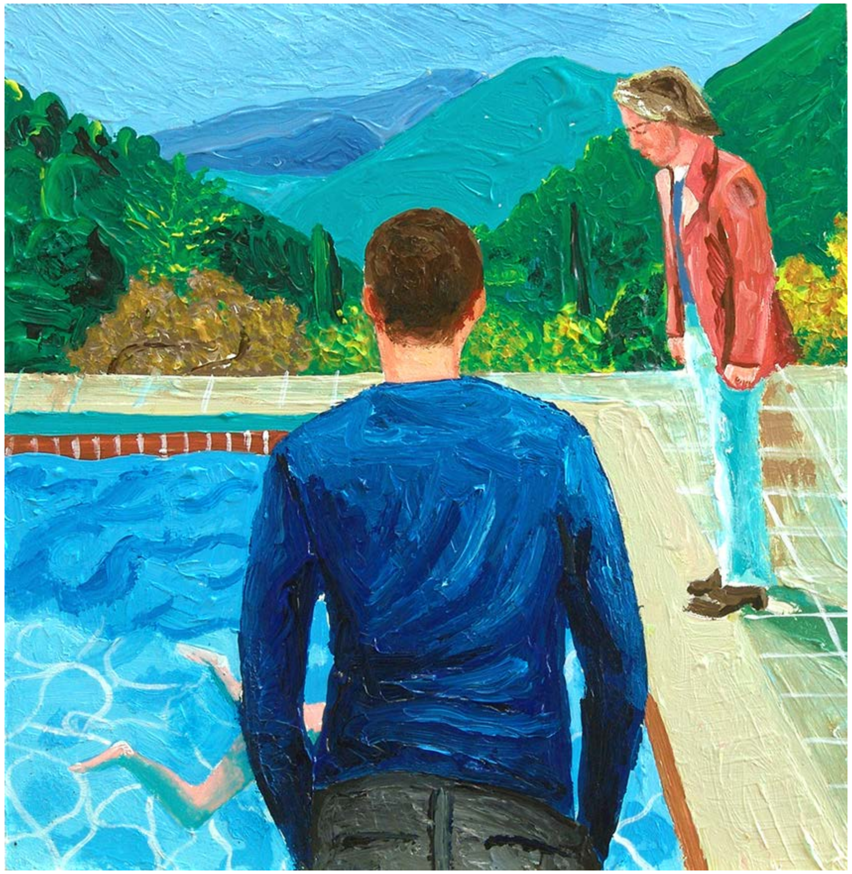
- *Suppose I own a swimming pool and I make a living by allowing others to use the pool for a fee (the pool is close on certain days and open only for certain hours)*
- *You figure out how to break into the pool undetected, and you break in and swim when the pool is closed*

An interesting analogy?

- *The act of swimming is not intrinsically wrong and swimming in the poll does not visible or physical harm to me, or to anyone else*
- *However, you are using my property without permission and it does not seem a sound justification for ignoring my property right if you claimed that you were hot and the swim in my pool made your life more tolerable*
- *Your argument would be no more convincing if you pointed out that you were not depriving me of revenues from renting the pool because you swan when the pool was closed*

More on the analogy

- Is the argument more convincing if instead of seeking your own comfort, you sought the comfort of your friend?
  - *Suppose you had a friend who was suffering greatly from the heat and so you, having the knowledge of how to break into the pool, broke in, in the name of altruism, and allowed your friend to swim while you watched to make sure I didn't appear*
- Is there **any moral difference** between **breaking into the pool** and **making a copy of PS**?



New rules

- In reality, instead of no rules, there are **new rules** some of which are **implicit** in the **technology**, and **others** are made **explicit**
- **Creative Commons** (Lessing 1999) to **encourage** and **facilitate** the **sharing of digital information**
  - CC makes available to anyone (for free) automatically generated licensing language and symbols so that users can label digital data they want to share
  - This permits users to allow or disallow commercial use, sampling, to require attribution, or charge for use, mixing and matching restrictions and permissions
  - For software, music, graphics, text, and presentations of all sorts
- **Technology** and notions of **right** and **wrong** are **intertwined**

References

- Lessing, L. (1999). *Code: And Other Laws of Cyberspace*, Basic Books
- Johnson, D. (2009). *Computer Ethics*, Forth Edition, Prentice-Hall
- Nissenbaum, H. (1995). “Should I copy my neighbor’s software?”, *Computer Ethics and Social Values*, Johnson, D. and Nissenbaum, H. (eds.), Prentice Hall, 201-213
- Stallman, R. (1995). “Why software should be free”. *Computer Ethics and Social Values*, Johnson, D. and Nissenbaum, H. (eds.), Prentice Hall, 190-199