

Open source model

in MOT strategy and S&T policies

Krzysztof Klincewicz

Graduate School of Innovation Management

 Tokyo Institute of Technology

Plan of presentation

- Definitions
- Open source software
 - history
 - innovativeness
 - role in commercial strategies and government policies
- Relevance of the open source model for other science and technology domains



Definitions

- **Proprietary standards** – technical specification and IPRs controlled by a company
- **Open standards** – can freely be used by everyone (*but often first created by companies*)
- **Public domain software** – anyone can do anything
- **Free software** – no price, but often proprietary
- **Open source software** – "*free as speech, not free as beer*": anyone can see and use the code, sometimes at a price
- **Copyleft** – uses copyright law to oblige licensees to ensure freedom for every user: anyone can use the code, but has to disclose any modifications to it (*defined in GNU GPL – GNU General Public License*)



GNU and GPL

- 1980 – Bayh-Doyle Act in the U.S. - technology transfer from universities
- 1984 – MIT transfers its software assets to AT&T
 - Richard Stallman leaves MIT Artificial Intelligence Lab and establishes GNU movement (*GNU = "GNU is Not Unix"*)
 - GNU General Public License (GPL) - the concept of "copyleft" is born
- 1998 – the term "open source" introduced to differentiate from "free software"



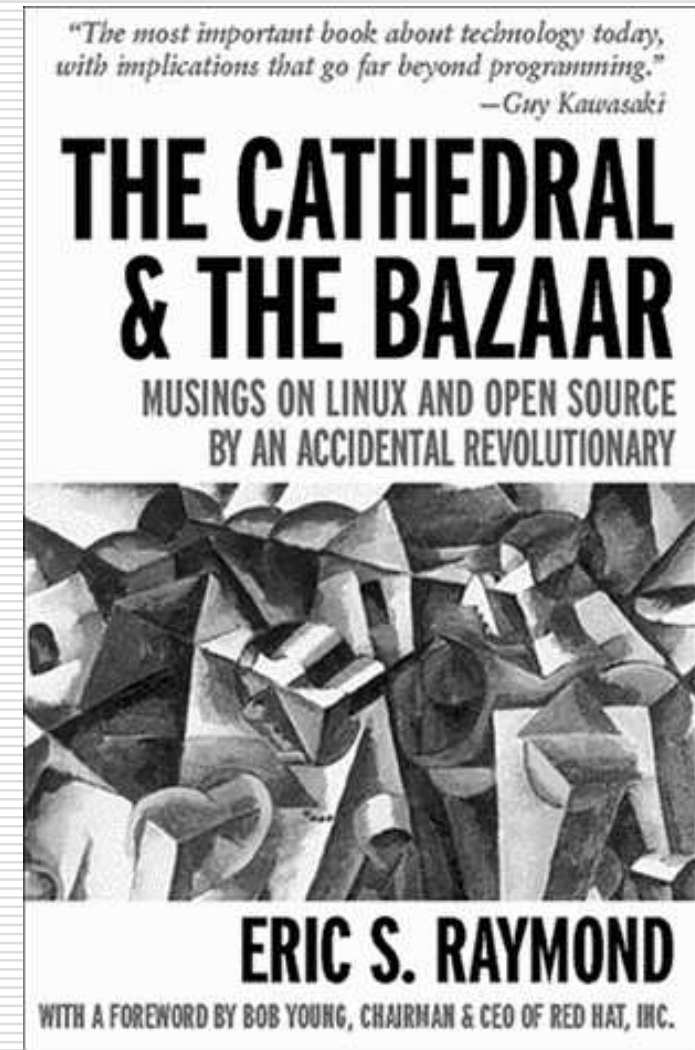
Linux

- 1991 – Linus Torvalds develops a prototype of Linux operating system
 - free and open replacement of Unix, which could be run on cheap computers
- Unique development model:
 - work of hobbyist developers
 - distributed community
 - motivation: fun, recognition, challenge
- Nowadays – established operating system, alternative for Windows, available for free download or purchase from various companies



"The Cathedral and the Bazaar"

- 1997 – Eric S. Raymond publishes "The Cathedral and the Bazaar", arguing that:
 - open source is an important process innovation
 - it offers a unique, "chaotic" software development model, which is more efficient than formalized corporate practices
 - it creates communities of developers and users, stimulating development and diffusion



"Cathedral" model of development



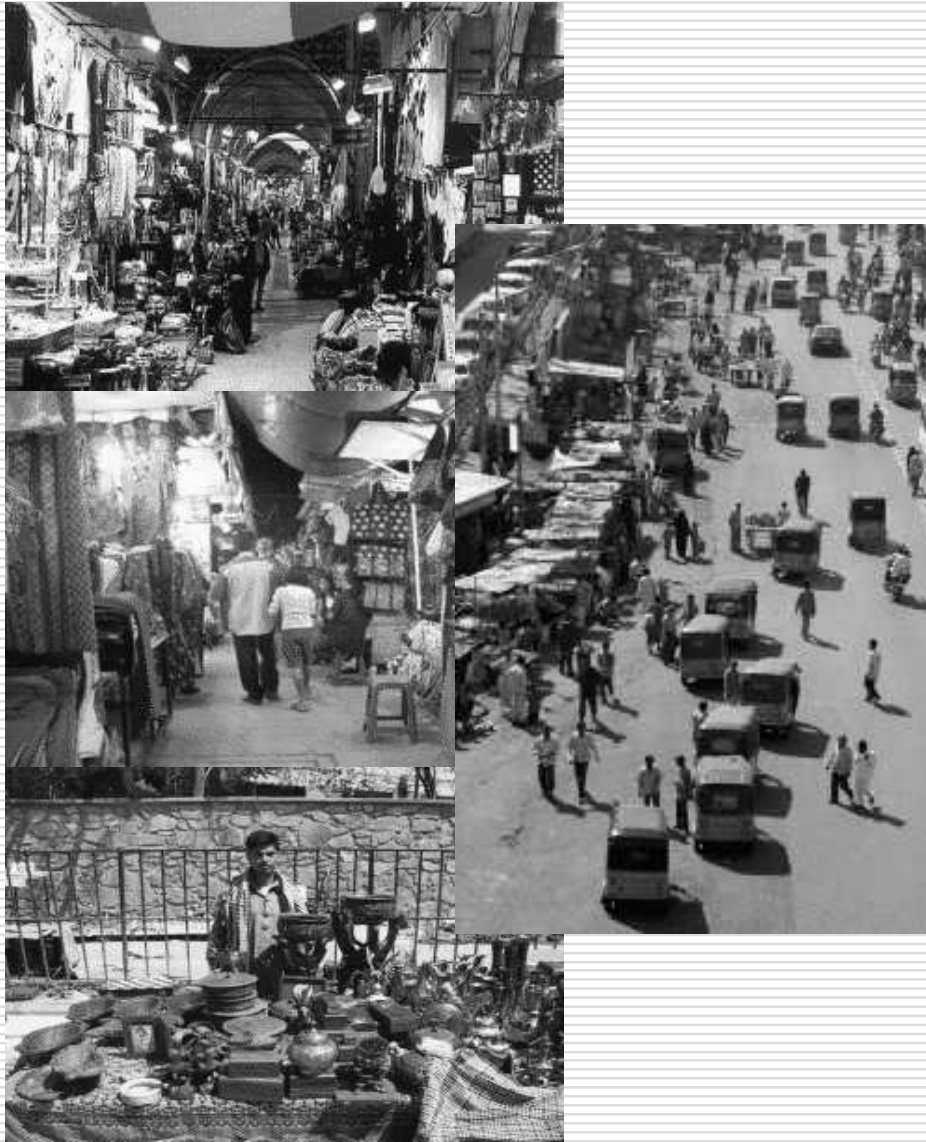
- large, ambitious, driven by long-term plans
- formal procedures
- hierarchy of roles

- limited responsiveness to user needs

- not benefiting from workgroups and direct communication



"Bazaar" model of development



- small-scale, focused on solving ad hoc problems
- chaotic, not based on procedures
- distributed work, using Internet to communicate
- roles resulting from experience and recognition by the community
- user intimacy and self-service



Open source as process innovation

- "Bazaar" model does not really exist - mature projects are formalized
 - hierarchies of roles (e.g. adding new code, deciding what should be included in next product version)
 - IT systems (*CVS – Concurrent Versioning System*)
 - long-term plans, marketing budgets and activities, technology and marketing alliances
 - investors, sponsors and employees
- but many elements were process innovations:
 - CVS and distributed software development
 - now internally used by companies including IBM



Open source as product innovation

- Analysis of 500 projects from open source portal Sourceforge <http://opensource.mit.edu/papers/klincewicz.pdf>
- Most are "me-too" products, often free versions of popular commercial applications
 - hobbyist developers seem to be more willing to participate in innovative projects
 - but it is difficult to promote new interesting ideas

	New technology	New for a platform	Existing technology
New market	Radical invention (breakthrough) 5 (1.0%)		Marketing innovation 3 (0.6%)
Existing market	Technology modification 4 (0.8%)	Platform modification 52 (10.4%)	No innovation 436 (87.2%)



Commercial open source (1)

- Software under GPL can also be sold – but source code of all modifications needs to be shared
- Red Hat Software
 - first IPO of open source company (1999)
 - now profitable
 - large enterprises prefer to pay for the software to get support guarantees
 - strategy of technology alliances – especially with hardware vendors and solution developers
 - problems with legal clones – e.g. CentOS



Commercial open source (2)

- Established commercial companies can benefit from using open source model:
 - "open" code of old applications – so that community of developers maintains them instead
 - use open source drivers and embedded software
- IBM – uses open source to fight Microsoft
 - Linux as the recommended enterprise platform
 - Source code of non-core technologies open
 - Protecting the OSS community with own patents
 - Focus on high-end hardware + proprietary enterprise applications -> *is IBM hypocritical?*



Commercial open source (3)

- New ventures based on open source model
- Developing own software but sharing it with developer community
 - balancing inhouse R&D and communities
 - more people testing and improving software
 - faster diffusion and customer self-service
 - double licensing
 - GPL for basic functionality, commercial license for advanced functionality
 - or paid licensors do not need to disclose source code of own modifications



Role of government

- Government initiatives to reduce dependence on proprietary standards
 - high royalties to Microsoft and other U.S. vendors
 - potential security threats
 - proprietary software cannot be easily modified
- Japan + China + South Korea
 - agreement between governments to develop regional Linux version – no results so far
 - Asianux – commercial initiatives of 3 companies
 - China: since 2003, government departments cannot upgrade or buy new foreign software, need to use with local substitutes (usually open source)



Science – open or privatized?

- Science traditionally public, open
 - but nowadays corporate R&D accounts for significant shares of scientific achievements
 - problems: withholding publications, restricting use by patenting
- Science becoming privatized, commercialized
 - Bayh-Doyle Act has also drawbacks
 - maybe governments should change the technology transfer legislation to preserve findings of publicly funded research for the public?



Problems with patents (1)

- August 2001 – John Keogh awarded patent by Australian Patent Office
- The invention was "transport facilitation device including a circular rim, a bearing in which a hollow cylindrical member is rotatable (...)"

(12) GRANTED INNOVATION PATENT	(11) Application No. AU 2001100012 A4
(19) AUSTRALIAN PATENT OFFICE	
(54) Title	Circular transportation facilitation device
(51) ² International Patent Classification(s)	B60B 001/00
(21) Application No. 2001100012	(22) Application Date: 2001.05.24
(45) Publication Date: 2001.08.02	
(71) Applicant(s)	John Keogh
(72) Inventor(s)	Keogh, John Michael
(74) Agent/Attorney	Sandercock Cowie 69 Robinson Street Dandenong Victoria AU

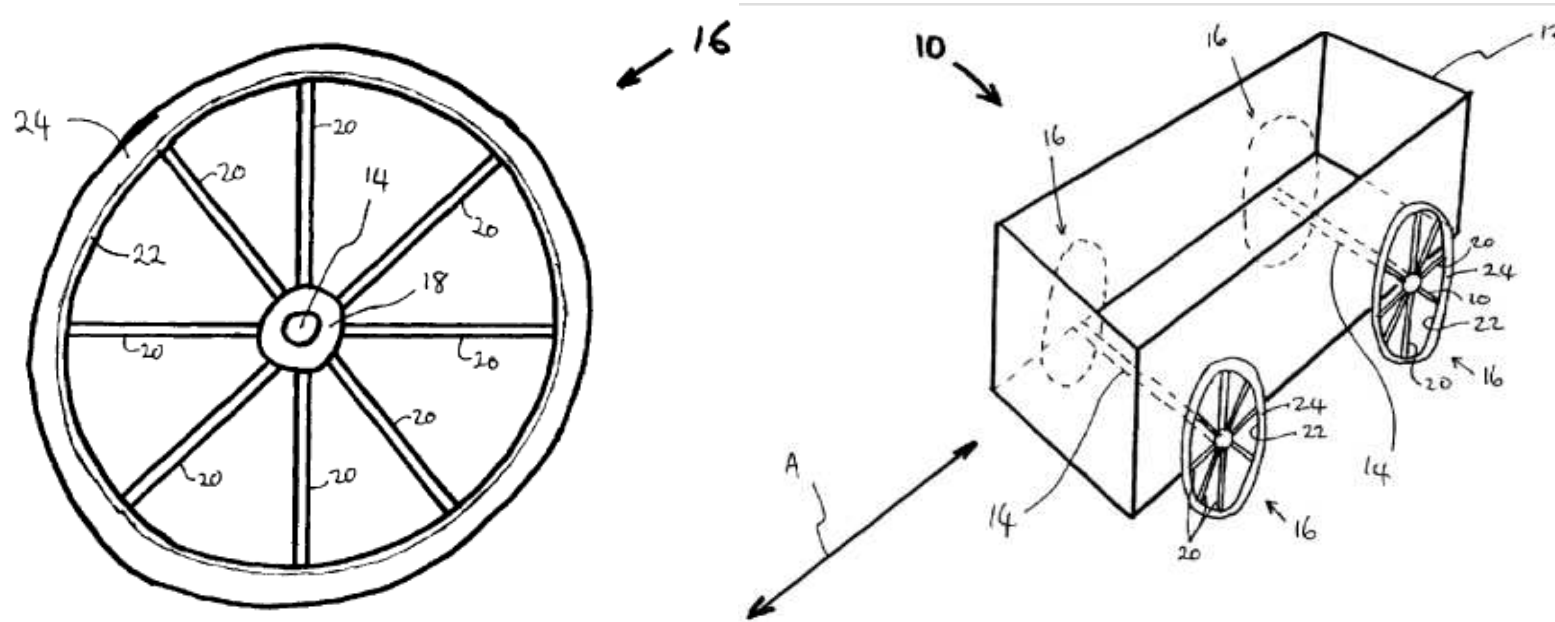


Problems with patents (2)

"In the past, transportation of goods and persons has been conducted in a number of ways. The predominant means has been transport of persons on foot, and carrying thereby of goods requiring transport.

Other means of transport have included, in colder climates, skis, sleds, toboggans and the like, which slide over a smooth surface (...).

Unfortunately, such smooth surfaces for sliding over are not generally available in warmer climates where snow and ice do not form naturally. As such, and in the absence of alternatives, foot transport may be required."



(John Keogh
promised not to
license fire)



Problems with patents (3)

- Problem with live-saving knowledge
 - receptor CCR5, used by HIV to attack the organism – patented by Human Genomics Services
 - 2 gens causing breast cancer – patented by Myriad Genetics
- USPTO awards patents for:
 - organisms and genes
 - intangible software algorithms (smileys like “:-”))
 - business methods (“how to squeeze more oranges in a box”)
 - designs (almost “normal” mobile phone)
- Companies believe the patenting system is inefficient



Open science

- But it does not mean that companies would not like to use patents to protect IPRs!
- Government initiatives - public research projects
 - e.g. Human Genome Project in 1990
- Community initiatives
 - peer-reviewed electronic magazines – competing with publishers such as Elsevier or Springer
 - Creative Commons license – similar to open source software: contents can be reused under certain conditions



Open source biotechnology

- Could be useful:
 - for developing countries (India and Africa "pirating" AIDS drugs)
 - when one company has too much control (Roche – Tamiflu suspected to cure bird flu)
- Problem: universities want to make money
 - "Columbia patents" – cotransformation technique used in rDNA, developed by prof. Richard Axel in 1970s, patented in 1983, annual revenues for Columbia University of 100 mln USD until 2000
- Problem: detailed knowledge may help terrorists
- Currently only hobbyist experimenting with open source licenses in biotechnology



Open source technology

- Tangible technologies (e.g. nanotechnology) require investments in infrastructure and manufacturing
 - difficult to establish communities of hobbyists
 - open source model better suitable for intangible goods: software, research findings, literature
- Some companies experimenting with the model:
 - Sun Microsystems – uses open source in software domain, but also opened detailed designs of certain computer chips so that everybody could manufacture them and stimulate diffusion of Sun's computer hardware platform



Questions?

Thank you!

Krzysztof Klincewicz

kklinc@esp.titech.ac.jp



Open source model in MOT and S&T polities

Krzysztof Klincewicz, Graduate School of Innovation Management, Tokyo Institute of Technology