

### **SERCI Annual Conference 2004**

# A bug's life: reductionism, holism and the open source *querelle*

Andrea Giannaccari and Andrea Renda Turin, 8 July 2004



### Introduction

Open source is frequently seen as a positive externality spurring from the intellectual endeavour of techies in search for reputation and intellectual challenge. Reality demistifies this view

#### Open source software has been subject to a heated debate over the past few years

- □ An intellectual commons?
- □ A new form of exchange or just free riding?
- A new technology frontier or a brake to disruptive innovation?
- □ The death of copyright?
- At government level, such a querelle seems to have been solved:
  - □ in favor of open source software...
    - ...to the disadvantage of proprietary software





# Three ways to look at software

#### 1. "stand-alone approach"

This approach implies that software is seen in isolation, as an information good. Competition authorities and governments usually adopt this "reductionist" view







# Three ways to look at software

#### 2. "value-chain approach"

If seen under a valuechain approach, software appears as a durable good, with many secondary markets







# Three ways to look at software

#### 3. "system-good approach"

Complexity and modularity in ICT markets determine the need for a more holistic approach, in which software is considered as a complementor in a system good







# The "stand-alone approach" (I)

**Proprietary software** 

- High Fixed Costs
  - $\square$  **R&D**

Proprietary Software is an information good, and as such is subject to a market failure. IP protection is needed for protecting valuable but risky investments

- □ In-house Testing
- Packaging
- □ Marketing
- Low or zero marginal costs
  - □ Re-production
- Appropriability
  - □ Free-riding/Emulation
  - Reverse engineering
  - □ Sharing/Piracy





# The "stand-alone approach" (II)

Open source software implies less initial fixed costs, and as such is less subject to market failure. Appropriation is part of the overall conception of open source software development **Open source software** 

- High Fixed Costs Low fixed costs

  - In-incuse testing Collective Testing
  - Packaging no packaging
  - Marketing no marketing
- Low or zero marginal costs
  - □ Re-production
- Appropriability Access to code
  - Free-riding/Emulation
  - Reverse engineering
  - Sharing/Piracy







# The "stand-alone approach" (III)

#### **Pricing information goods**

Because of low marginal costs, price discrimination is an optimal pricing strategy for information goods





8



# The "stand-alone approach" (IV)

#### **Bundling and versioning information goods**

Firms can use versioning and bundling strategies in order to efficiently segment consumers in different groups with different willingness to pay. Producer surplus and incentives to invest in R&D are maximized







# The "stand-alone approach" (V)

#### Pricing open source software

With open source software, price equals marginal costs. Consumer surplus is maximized under a static analysis, but what about dynamic efficiency?







# The "stand-alone approach" (VI)

#### OSS v. Proprietary software: static reductionist analysis

	OS software	Proprietary software	winner
Price	little or no charge	price above marginal cost	OSS
Consumer surplus	maximized	minimized	OSS
Deadweight loss	None	Small	OSS
R&D	no need?	in-house investments	depends
Quality	Collective testing	in-house testing	depends
Security	more attack, more defence	need for corrections - alpha and beta testers	depends





# The "value-chain" approach (I)

#### The marketing of software spurs demand for IT services





Source: Berlecon Research



# The "value-chain" approach (II)

#### Two business models



#### Monopoly

- Two-sided market
- R&D investment
- Building a network of ISO
- Building customer base
- Competition "for" the market under network fx



#### Competition

#### Non-market

- □ Small sunk investments
- □ Supply of OSS
- □ Bundling with IT services
- Relational contracting





Competition

#### **Beware of geeks bearing gifts!**



# A system-good approach (I)

Under a systemgood approach, OSS is seen as a complementor in a wider system good. This software is often used to convince end users to purchase other complementors Software is usually included in a system good: all complementors are necessary for final users



Increasingly, OSS-based firms bundle OSS with other (proprietary) software or with hardware complementors





# A system-good approach (II)

Intra-system competition depends on platform vendor's choice. Competition authorities should not promote open source by mandating interoperability between proprietary software and OSS Different system designs involve different degrees of competition in the market





All system architectures have advantages and disadvantages. A general right to interoperability would preclude the choice of a closed or a semi-open system...



# A system-good approach (III)

In markets with strong network externalities, platform vendors compete for the market rather than in the market.







# A system-good approach (IV)

p

pc

C

Dc

MR

In a non-proprietary world, firms do not engage in competition "for" the market, but only compete "in" the market with diversification strategies In a non-proprietary world, end users do not enjoy the benefits of standardization, firms do not invest in R&D and only smallscale, path-dependent innovation takes place



Law and Economics Lab

 $\mathbf{Q}$ 



# **Competition policy issues**

In most cases competition authorities faced a glamorous impasse when dealing with software. More complex approaches can help trustubusters get it right



Value-chain approach
 Calculation of market shares
 Bundling in aftermarkets
 Abuse of economic dependency

# System-good approach

- Market definition/barriers to entry
- Technological tying/bundling
- Dynamic efficiency
- No mandatory interoperability, no compulsory licensing



### The public boost towards OSS

Many governments are making efforts to promote free or open source software Government subsidies of R&D for OSS

- Standardization on using OSS
- Procurement preferences for OSS





# European initiatives towards OSS

A massive political incentive has also been given by EU Institutions

#### *European Parliament resolution 2001/2098:*

the Commission and Member States have "to promote software projects whose source text is made public".

#### Commission IDA program:

"the software is still not extensively used in most of the European Member States' public administrations" but "on general-purpose servers as well as on office desktop, Open Source software will present tomorrow the most realistic, and sometimes the only real technical and economical alternative to Microsoft products"





# European initiatives (follows)

eEurope 2005 Action Plan and the European Interoperability Framework

#### eEurope 2005 Action Plan: the Commission has to

"issue an agreed interoperability framework to support the delivery of pan-European e-government services to citizens and enterprises. (...) It will be based on open standards and encourage the use of open source software; (...) it also intends to support standardisation with a view to wider use of open standards and open source software"

#### European Interoperability Framework:

"OSS corresponds to the objectives of this Framework and should be assessed and considered favourably alongside proprietary alternatives"





#### EU Governments reactions (examples)

Many political initiatives have tried to foster the open source movement and to spread OSS use in PAs **Germany- Bundestux initiative:** 

"the introduction of a free operation system in the Bundestag would be necessary to promote basic regulation, competition and location policy, as well as for democratic reasons"

#### **Government-IBM-SuSE** agreement

French Agency for Technologies of Information and Communication in Administration

*"encourage administrations to use free software and open standards"* 





### Proprietary solutions vs OSS

Standardized software and open source software are mirror images

- Proprietary software (customer focused)
  - generate revenue by selling or licensing their software (exploit IPRs)
  - identify market needs
  - link product development closely to market demand
- OS software (developer focused)
  - developers usually volunteer
  - less concerned about market demand
  - 'by techies for techies'





### Is there a market failure?

Is it necessary to require a government intervention in the software market?

- From 1988 to 2000, revenues from worldwide proprietary software increased from \$ 35 billion to \$ 171 billion
- Unconcentrated market: in 2002, the four largest firms in the proprietary software industry accounted for 26.7% of total revenues
- The HHI for the software industry was 244
- In 2001, worldwide output was more than 20 times as large as it was 12 years earlier
- From 1997 to 2001, the software CPI fell by 20.5% while the CPI for all items rose by 9.5%. The real price fell by approximately 27.4%
- Turnover: five of the top ten companies in 1990 did not make the list in 2000





#### The Economist: OSS is "for love, not money"

There are different business models associated to OSS Sell complementary software

Sell complementary hardware

Sell complementary services

 (assistance, training, support)





### Is OSS more cost-effective?

The software costs for business are usually measured on a "total cost of ownership" (TCO)  Software acquisition costs (less than 5%)

**TCO of IT systems** 

- Cost to customize the system to user needs
- Cost of maintenance and support
  Training costs





#### Which system is more cost-effective? It depends on the use: products should largely be considered on their merits for the project at hand...

 $\dots \mathsf{TCO}$ 

#### META Group, 2001

"Linux is typically not a low-cost alternative [as compared to Microsoft Windows] when viewed from a total cost-of-ownership perspective, because it costs more for organizations to support it"

Other studies by IDC, Forrester Research and Giga Research have found similar results...





A light-handed regulation?

The software industry is not perfect, no industry is. But this not represents a market failure There are imperfections in the market, not market failure: some socially beneficial transactions do not occur in traditional market

Open source software is itself a private means of remedying some of these market imperfections





Both approaches, and business models, have advantages and disadvantages. **Case-by-case evaluation** 

- It is necessary to evaluate OSS and proprietary software on a case-by-case, product-by-product basis
- In some cases, OSS is better than proprietary software with regard to price, technical advantages, or both.
- In other cases, proprietary software may be the best choice because its strengths outweigh the fact that OSS provides the source code for free





A sound government intervention in the market should be aimed at funding R&D of software Subsidize basic research

It is widely agreed that for-profit firms do not have sufficient incentives to produce research to the point where benefits—public and private— equal private costs

"viral nature": the GPL effectively prevents profit-making firms from using any of the code since all derivative products must also be distributed under the GPL license





#### Subsidize basic research (follows)

Basic research is a public good with strong positive external effects that will not be provided by the market It is unclear what the results of basic research are going to be, how long it will take to find them, and what they may eventually be used for.

- Potential positive spillover effects of basic research are widespread and very difficult to internalize by commercial companies,
- therefore they have little financial incentive to engage in basic research.
- Basic research helps to promote new scientific developments that could be the basis for new products, whether proprietary or open source





In the last 20 years governments have shown no particular skill in choosing industries to support as part of 'industrial policy' initiatives

#### Bad track records for govs

- Whether "open code" in any given situation is "as powerful" as "closed code" is an everyday business judgment that should be made by businesses, governments, and private users
- Who 'wins' in the market is not a policy issue that has to be decided by bureaucrats or legislators, or even by lawyers and economists
- Sound economic analysis is needed in order to assess the relative strength of OSS and proprietary software on a caseby-case basis





# www.law-economics.net