Open Source Software Usage in the Schools

conceptual strategy

Kliment Simoncev
simoncev@gmail.com
open source advocate
on behalf of MoES

Drafting conceptual strategy

- Leverage ***open source*** *software* in different use cases in education
- Enabling ***knowledge sharing*** and collaboration between teachers and students
- ***Unveil myths*** around open source software
- Introduce concept of ***open education***
- Analysis and identification of ***common open source applications*** that can be useful in particular context of educational curriculum
- Provide proposal for ***sustainability model*** of open source

Key terminology

- **Open Source** software – OSI (Open Source Initiative)
- **Free** software – FSF (Free Software Foundation)
- Open **standards –** W3C, Oasis, etc.
- Open **systems** - POSIX
- F/OSS (or **FLOSS**) – inclusive term, used publicly by EC , 2000.


Very short history

- “Free software” is introduced by software development for the GNU operating system began in January **1984,** and Free Software Foundation (FSF) was founded in October 1985
- “Open-source software" was proposed in **1998** as a replacement label for "free software". Later that year, Open Source Initiative was founded to promote the term as part of "a marketing program for free software"

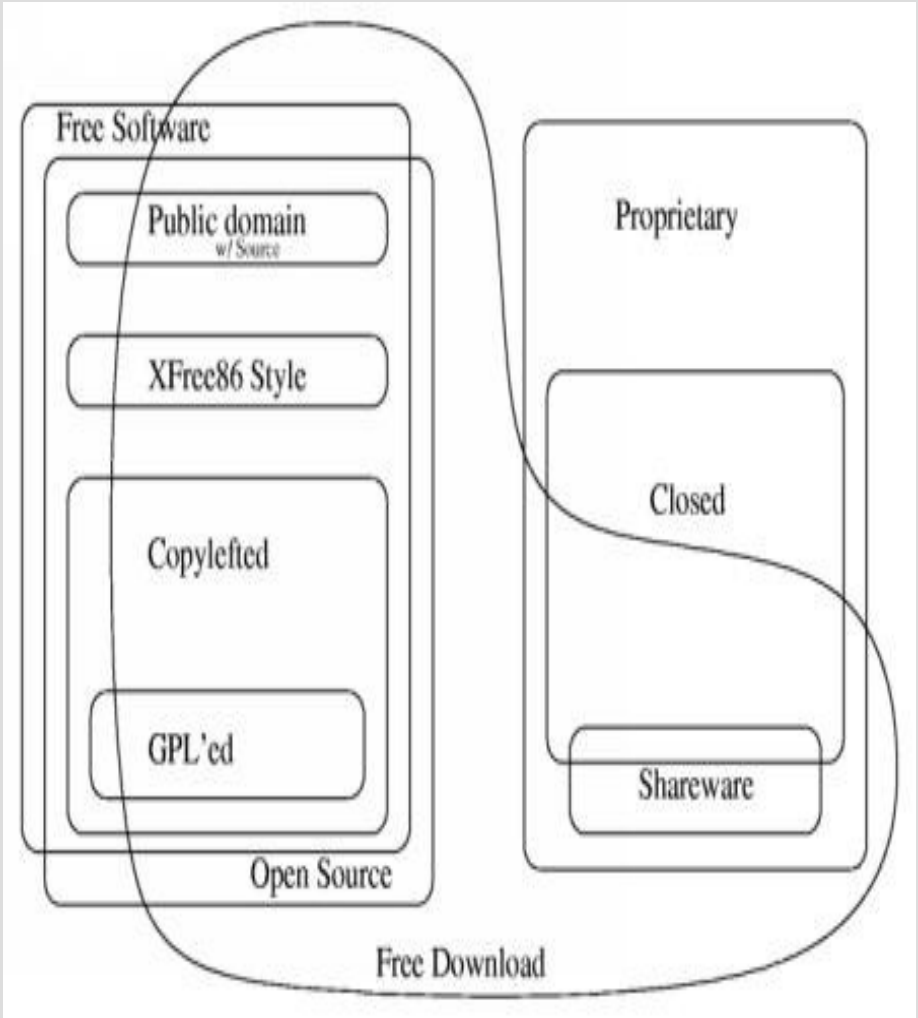
Free vs. open source

- Free Software Foundation
- Richard Stallman
- **Gnu Public License**: 'viral', ensuring; openness of code in perpetuity
- Freedom of information
- Leftist; communitarian; idealist

- Open Source Initiative
- Eric Raymond
- **BSD-style licenses**: not placing any restrictions on use of software
- Better quality software
- Libertarian; pragmatist

The term FLOSS (FOSS,F/OSS) is often used *to bridge the ideological divide* between the free software and open source software movements. It can also be used as a *neutral term* when discussing free / open source software with those of differing ideological viewpoints.

Software licenses



- **Freedom 0**: The freedom to run the program, for any purpose
- **Freedom 1**: The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.
- **Freedom 2**: The freedom to redistribute copies so you can help your neighbor
- **Freedom 3**: The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

OSI Licenses

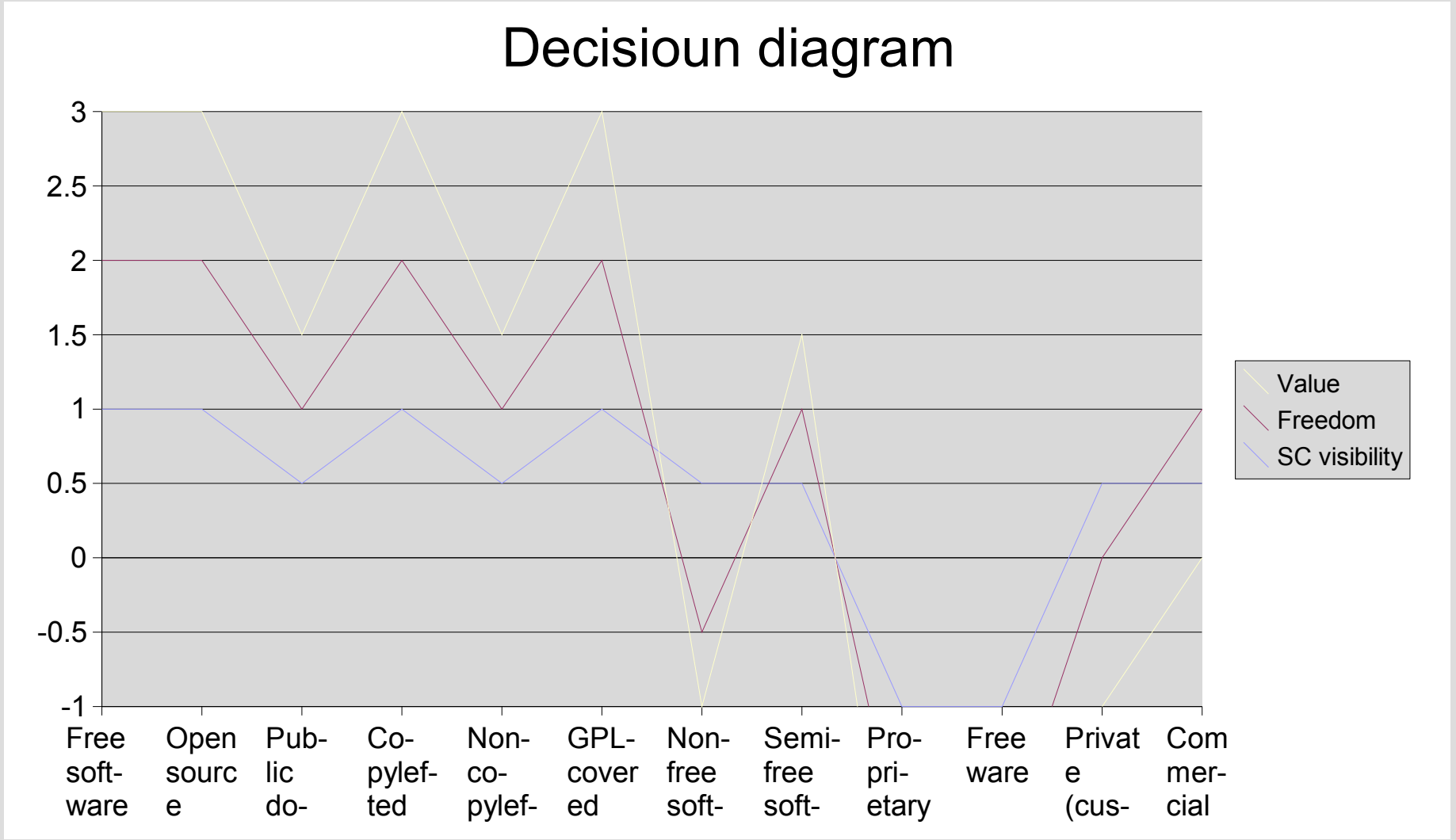
Mozilla Public License	• Firefox • Thunderbird
Sun Public License[1]	• NetBeans
Eclipse Public License	• Eclipse
GPL/LGPL License	• Lazarus IDE – free pascal • free pascal • GNU Compiler Collection • AbiWord • Open Office • Celestia • 7-Zip • Gimp • Mplayer • Blender • Inkscape • Wine • Kgeography • Gaim • Kig • Kino

- A lot of choices!
- Famous:
  - GPL/LGPL
  - Mozilla
  - Apache
  - BSD
  - MIT

Software categories – personal quantification (just as example)

Category	*SC visibility*	*Freedom*	*Value*
Free software	1	1	1
Open source	1	1	1
Public domain software	0.5	0.5	0.5
Copylefted software	1	1	1
Non-copylefted free software	0.5	0.5	0.5
GPL-covered software	1	1	1
Non-free software	0.5	-1	-0.5
Semi-free software	0.5	0.5	0.5
Proprietary software	-1	-1	-1
Freeware	-1	-1	0
Private (custom) software	0.5	-0.5	-1
Commercial software	0.5	0.5	-1

Software categories – speculative diagram



Benefits of F/OSS

- Social, economic and cultural aspects
  - F/OSS supports the local IT industry and digital self-sufficiency
  - F/OSS supports entrepreneurship and business formation
  - F/OSS supports innovation, local solutions and learning
  - F/OSS supports local content creation and consumption
  - F/OSS reduces vendor dependence and lock-in
  - Open source *raises the profile of developing countries* in the global economy
  - F/OSS puts user needs first: F/OSS shifts the competitive advantage
  - Open source promotes transparency and accountable government
- Transparency
  - Knowledge sharing
  - Collaboration
- Creativity
  - Do not guarantee creativity by itself, but it can make creativity run and work in unrestricted way.

Benefits of F/OSS in Education

- Greater **learning of concepts** rather than products – new approaches of teaching and learning
- Possibility of **localization** (translation)
  - promotion of *cultural identity*
- **Customization** and personalization
- Encourage **Innovations**
  - *Stimulating self-promotion*
- Lower total cost of ownership (**TCO**)
- **Bridges the digital divide** and ensuring affordable access to ICT
  - *equality principle* & affordable access to ICT as *democratic right*
- Alternative to Illegal Copying

Pitfalls of FLOSS

- *Lack of awareness of F/OSS*
- *Poor Internet and international links*
- *Software piracy , so there is no clear, general evidence for decision makers*
- *However, there is multiple path to resolve that issues*

F/OSS in education

- **F/OSS in IT curriculum**
  - Programming languages
  - Databases
  - Learning concepts
- **F/OSS as education tool in non-IT curriculum**
  - **New teaching and learning methodology tools**
    - **e-learning systems**
    - **interactive courses**
    - **applied programming in natural science curriculum – virtual lab**
  - **Multidisciplinary projects**
  - **Edutainment (games)**
  - **Art**
- **School administration**
- **ICT infrastructure**

F/OSS primer

- Educational technology
  - Content Management Systems
  - Learning Management Systems
  - Interactive virtual learning systems
  - Desktop – end user – learning workstations
  - Server – Web centric – centralized infrastructure
- Collaborating communities of practice (CoP)
  - Domain
  - Community
  - Practice
  - NOT just a web site!
- Open Content – *Creative Commons License*
- Free Operating system – Linux

creative commons

Debian GNU/Linux

Who will use F/OSS ?
Q&?

- **Q1**. To what **degree** would students and teachers want to or be able to become involved in co-creating digital resources?
- **Q2**. Are teachers and students **capable** of maintaining theirs own open system infrastructure?
- **Q3**. What kinds of F/OSS systems would teachers and learners want and what levels of **functionality and ease-of-use** would they require?
- **Q4**. What kinds of **authoring tools** need to be customized/adapted/developed in order to support teachers and learners to become more actively involved in co-creation of resources?
- **Q5**. What kinds of resources and **policies** do we need to put in place in order for the different communities (F/OSS, software industry, researcher and educators) to work together?
- **Q6**. What **business models** would need to be developed to enable co-creation to support all sectors?

FLOSS participation

Level/Role	**Spectrum of roles from less to more participation and structuralization**							
8 (highest)	Passive User	Using software as most of commercial software						
7		Reader	Peer reviewing, learn from source					
6			Bug Reporter	Discovering reporting bugs. May not read source code, but in most cases do.				
5				Bug Fixer	Understand small portion of source code and change it.			
4					Peripheral developer	Improvements and small changes of software		
3						Active developer	Active development	
2							Core Member	Grounding and leading
1 (deepest)								Project Leader
ROLE description	Less participation, end-user	Understanding, Reading	Testing, Reporting	Bug fixing	Occasionally development	Major new development	High constant participation in long-run, development	High participation, responsibility

How to make F/OSS sustainable?

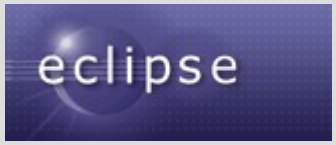
- **Awareness** for F/OSS – in general
  - Monitoring development of F/OSS – **stay in-touch with community**
  - **Collaborate** with developers
  - Education and **documentation**
  - Develop **local resources**
  - Translation of software as **continuous process**
  - Customization and **extension** of software
  - Development of brand **new products**
  - Leverage **open standards**
  - **Warranty services** and service level agreements
  - Different sw. distribution channels, use **portable** software

Showcase 1 – Lazarus IDE

- Translation in 2 languages.
- Compilation of source code on Ubuntu 6.06 platform.
- Installation for Windows.
- Debian distribution format.
- Compilation of Lazarus developed OSS project (lazix) included in distro.
- Testing free pascal compliance to standard secondary school pascal curriculum.

Showcase 2 - Physics – programming & science – Eclipse development

- Integrating all simulation programs into Eclipse run-time frameworks as plugins
- All-in-one application for physics simulations
- Customizing development IDE as branded product.



Showcase 3- Interactive chemistry - JMol

- Presentation of molecular models directly from web site running Jmol applet.
- Develop teaching tutorials based on Jmol.
- Using already available web sites on internet running chemistry systems based on Jmol.

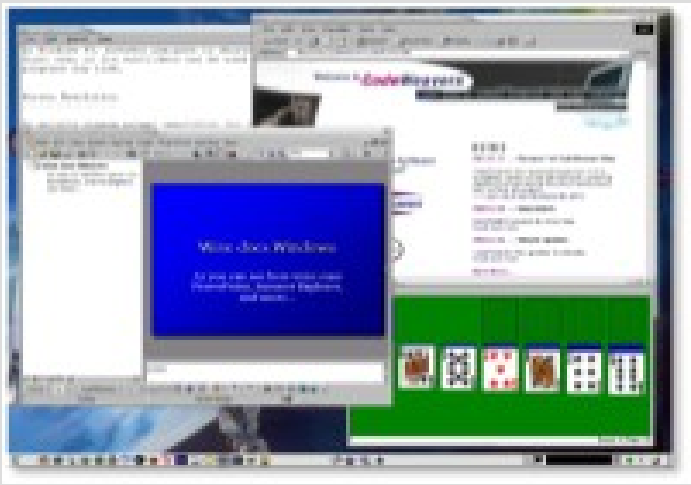
1. Moodle and Jmol integration: http://docs.moodle.org/en/Jmol_filter
2. Teaching web application: http://www.stolaf.edu/depts/chemistry/mo/struc/
3. Gallery of molecules: http://molvis.sdsc.edu/fgij/gallery.htm
4. Molecules In Motion: http://www.moleculesinmotion.com/
5. The Virtual Museum of Minerals and Molecules : http://virtual-museum.soils.wisc.edu/

Showcase 4 - Open Office – extending with macros & components

- Customization and personalization by extending OpenOffice with macros.
- Use already developed macros (available as open source projects) as start up project for your own macro development.
- **Automation of Office Administration work** with customized open source office productivity suite.

Showcase 5 -Wine

- Running Windows applications on Linux with help of Wine.
- Tool for having only one educational OS for running application not available on Linux.
- Portability



Showcase6 -Apache web server running PHP applications - phpBB

- Showcase of PHP technology and LAMP (Linux/Apache/MySQL/PHP) technology stack.
- **Apache** as most popular and used open source web server
- **phpBB** is a popular free and open source forum system written in the PHP programming language
- Compatibility with multiple database management systems.
- A large community of users providing free support and customizations.

Showcase 7 -Remastering of Linux distros

- Re-mastering of official Ubuntu distribution – installation and live CD.
- Branding of distribution: educational look & feel.
- Distribution of selected desktop software for education purposes
- *Starting point for more specialized* Linux distributions for special purposes

Conclusions

- F/OSS communities control the ***developmental dynamic of evolving* good**
- More about ***open future*** (and opportunity) than about access to currently existing source-code text.
- **Preferable choice**, especially in situation of new economy model promoting by F/OSS philosophy, proving itself as viable by economic criteria and still keeping concept of freedom and openness as strategic objective
- Try to make some ideas and concepts behind **F/OSS phenomenon** more clear
- Directions toward practical implementations of F/OSS concepts in education

Thank You

- Any questions?