Open source perspectives from the UK

Sebastian Rahtz

OSS Watch

Outline

1 OSS Watch

2 Current issues facing OSS Watch

3 UK national policies

4 The JISC draft policy

5 A University Open Source Policy?

6 Conclusion

OSS Watch

Open source
perspectives
from the UK

Sebastian
Rahtz

OSS Watch

Current issues
facing OSS
Watch

UK national
policies

The JISC draft
policy

A University
Open Source
Policy?

Conclusion

I am Sebastian Rahtz:

- Information Manager for *Oxford University Computing* Services
- Manager of JISC's *OSS Watch*, a UK national advisory service

sebastian.rahtz@oucs.ox.ac.uk

- Give an overview of what *OSS Watch* does
- Highlight the common issues we encounter
- Outline some recent UK government work
- Discuss what a university open source policy might say

OSS Watch

- The JISC (Joint Information Systems Committee) coordinates educational IT structures in the UK. Directly funded by the state at the same level as research councils.
- JISC runs the physical network for HE and FE, provides services, and funds applied research.
- OSS Watch is funded from 2003-2006 as a UK national Open Source Advisory Service.
- OSS Watch has 3.25 FTE based in Oxford University Computing Services' *Research Technologies* section.

OSS Watch provides unbiased advice and guidance about free and open source software for UK further and higher education. OSS Watch is not set up to be an advocacy group.

- Most universities deploy backoffice open source software
- Many FE colleges have open source at the grass roots but not institutionally
- Educational software like VLEs are on everyone's agenda
- No-one has addressed staff involvement in open source at a policy level
- The JISC is very much in favour of open source
- The UK government has fine words but not much action

- Most governments have put studies of open source in place (eg France, Denmark, Germany, Spain, Sweden, Finland, UK)
- Germany has seen a trial involving the GPL
- Spain has a provincial government deploying OSS widely
- The EU has agreed to software patents via an incredibly undemocratic process. An interesting saga.

Briefing material	2-3 page guidance notes
Conferences	Twice yearly
Project support	Face to face discussion
Reports to JISC	JISC open source policy
Collaboration	Working with other groups
FE roadshows	Regular 1 day intros
Demonstration Software	Knoppix and Open CD
Survey	Autumn 2004, Autumn 2006
Website	(naturally)
Workshops	Focused small events
International Conference	March 2006

Recent Briefing Notes

- *What is version control? Why is it important for due diligence?*
- *Where to go to try out some open source software*
- *5 essential open source tools for the sysadmin*
- *Linux Standards Base: what is it and why is it important?*
- *OSS Watch KNOPPIX 3.6*
- *Dual Licensing: A threat to Open Source Software?*
- *Standards and Open Source*
- *Assessing the Role of Libre/Open Source Software for European Industry, The Hague, 19 November 2004*
- *Open Standards and Libre Software in Government, The Hague, 18 November 2004*
- *4th International System Administration and Network Engineering Conference, 30 September 2004*

January 20th 2004: 'Open Source: national frameworks'

Alan Robiette: *developing JISC open source policy and practice*

Marc Bressers: *OSOSS - open source and open standards in The Netherlands*

Rishab Ghosh: *open source across Europe*

David Casal: *open source business across Europe*

David Rayers: *open source and the BBC*

Michael Coen: *investing in proprietary, in-house or open source software or services*

■ ■ ■

How to choose the right licence?

- It is hard to know at the start of a project which licence to use
- The GPL scares off industrial collaborators
- The LGPL is hard to understand
- BSD-like licences seem to invite other people to not play the game
- Creative Commons is unpleasantly attractive because it smells like open source but lets you be restrictive

Note the anti-GPL noises coming from SAKAI.

Worries about:

- *Speed of change:* who knows where this stuff is going?
- *No roadmap*: how do we plan our support needs?
- *Functional gaps*: will we need support to plug them?
- *Licensing issues*: does this impact on who can provide support?
- *ISV endorsements*: will anyone admit to being able to help us?

Building communities

Projects *tend* to start small, and often go in one of seven directions:

- stay small: remains a nerd tool
- gather users but no new developers: frustrated users
- fragment when primary leader loses interest: unattractive for new people
- develop power but with minimal documentation: no way to find the power
- grow within an expert community: high price for admission
- go commercial: stops being 'free'
- simply die

What we aspire to

- Apache: technical strategy, formal democratic management, enviable reputation
- Docbook: described in all the books in the shop about XML
- Moodle: tops the rankings in Google
- Firefox: smooth as butter install, reference implementation
- uPortal: shared development between academia and business

The visionary has the Big Idea, makes the long-term
  decisions

The leader makes the medium-term decisions

The programmer implements the functionality and makes
  the short-term decisions

The tester finds the bugs

The apprentice programmer fixes the bugs

The documentor write the manual

The communicator tells other people how good it all is

The distributor packages it up for new users to try

How many of these roles can safely be filled by one person?

■   ■   ■

Why do we get involved in open source?

As either creators or consumers, we have a variety of
motives for doing open source:

- to save money (*of course*)
- to share work (*enlightened self-interest*)
- because it works (*software engineering*)
- to learn (*apprentice work*)
- for fun (*someone has to find it fun*)
- for social justice (*break the capitalist system*)

but not very often because we are told to.

It is not unreasonable for government to consider:

- ways of saving money in public procurement
- trying to avoid duplicate effort
- promoting efficient development of IT systems
- allowing for skill development

Whether government has an agenda for fun or social justice is less clear!

www.ogc.gov.uk/index.asp?docid=2190#finalreport

Viability of OSS Open Source software is a viable and
credible alternative to proprietary software for
infrastructure implementations;

Obstacles to implementation ... for desktop applications,
the current lack of complex functionality which
can affect ease of migration and
interoperability ...

Costs and benefits ... can generate significant savings in
hardware and software costs for infrastructure
implementation, and reduce the licensing costs
and hardware refresh requirements for desktop
implementation.

Background 2: Recommendations from open source trials

Public sector bodies should:

- examine carefully the technical and business case for implementation of Open Source software and the role which OSS could play in current and future projects, working with their outsourced IT providers where appropriate;
- review the potential for server consolidation, comparing the benefits of OSS with proprietary solutions;
- consider the potential costs and benefits of migration to an OSS desktop for transaction users, (potentially in conjunction with use of 'thin client' architecture solutions);
- identify the role of open standards in future IS/IT strategy and policy, in conformance with the e-Government Interoperability Framework (eGIF);

- consider requirements for the development of skills in Open Source development, deployment and operation within the organisation, and review the availability of such skills in their outsourced IT service providers;
- review their current infrastructure and applications — in collaboration with their outsourced IT providers where relevant — well in advance of any planned procurement or renewal, and determine whether current technologies and IT policies inhibit future choice; and if so consider what steps may be necessary to prevent future 'lock in'
- consider the benefits of incremental change by diversifying OSS use beyond the server platform to products like Email, LDAP, Web and internet Browser.

(verbatim from eGU)

http://www.govtalk.gov.uk/schemasstandards/
egif.asp

- The e-GIF defines the technical policies and specifications governing information flows across government and the public sector.
- They cover interconnectivity, data integration, e-services access and content management.
- Version 6.0 contains the high level policy statements, management, implementation and compliance regimes, whilst technical policies and specifications are contained in the Technical Standards Catalogue (TSC).

Justification for the UK government policy

- . . . always procure a solution that gives value for money.
- There is a need to ensure that interoperability of systems is provided and maintained.
- Every effort should be made to reduce the cost and risk to government systems. . . by:
    - acquiring best value for money solutions
    - removing the reliance on individual IT suppliers
    - providing more flexibility in the development, enhancement and integration of systems
    - vesting the ownership of bespoke and tailored software code with Government where this offers value for money.
- Security of government systems is vital.
- There is a need to maximise returns on and benefits from public investment in publicly funded R/D software.

(verbatim from OSS Policy)

http://www.govtalk.gov.uk/policydocs/
consult_subject_document.asp?docnum=905

Version 2 28 October 2004

The key decisions:

- UK Government will consider OSS solutions alongside proprietary ones in IT procurements. Contracts will be awarded on a value for money basis.
- UK Government will only use products for interoperability that support open standards and specifications in all future IT developments.
- UK Government will seek to avoid lock-in to proprietary IT products and services.
- Publicly funded R/D projects which aim to produce software outputs shall specify a proposed software exploitation route at the start of the project. At the completion of the project, the software shall be exploited either commercially or within an academic community or as OSS.

(verbatim from OSS Policy)

The key decisions:

- UK Government will consider OSS solutions alongside proprietary ones in IT procurements. Contracts will be awarded on a value for money basis.
- UK Government will only use products for interoperability that support open standards and specifications in all future IT developments.
- UK Government will seek to avoid lock-in to proprietary IT products and services.
- Publicly funded R/D projects which aim to produce software outputs shall specify a proposed software exploitation route at the start of the project. At the completion of the project, the software shall be exploited either commercially or within an academic community or as OSS.

(verbatim from OSS Policy)

The key decisions:

- UK Government will consider OSS solutions alongside proprietary ones in IT procurements. Contracts will be awarded on a value for money basis.
- UK Government will only use products for interoperability that support open standards and specifications in all future IT developments.
- UK Government will seek to avoid lock-in to proprietary IT products and services.
- Publicly funded R/D projects which aim to produce software outputs shall specify a proposed software exploitation route at the start of the project. At the completion of the project, the software shall be exploited either commercially or within an academic community or as OSS.

(verbatim from OSS Policy)

The key decisions:

- UK Government will consider OSS solutions alongside proprietary ones in IT procurements. Contracts will be awarded on a value for money basis.
- UK Government will only use products for interoperability that support open standards and specifications in all future IT developments.
- UK Government will seek to avoid lock-in to proprietary IT products and services.
- Publicly funded R/D projects which aim to produce software outputs shall specify a proposed software exploitation route at the start of the project. At the completion of the project, the software shall be exploited either commercially or within an academic community or as OSS.

(verbatim from OSS Policy)

'The policy on exploiting R/D software will not apply to
software developed in the areas of defence, national
security or law enforcement. It will also not apply to software
developed by Trading Funds.'

- DTI, eGU and JISC will disseminate information on the distinct types of OSI compliant licences to support use, development and exploitation of OSS by government organisations and publicly funded R/D teams.
- DTI will include the R/D software exploitation policy in guidance on collaboration agreements. Research Councils will include the R/D software exploitation policy in guidance on research grants and contracts.
- DTI, Research Councils and JISC will explore the feasibility of providing unified access to publicly funded R/D OSS.
- eGU will explore with Government, industry and other stakeholders further activities to support OSS use in the public sector.

(verbatim from OSS Policy)

- How was this policy derived?
- Is this a law or a nice idea?
- How does it affect me in UK HE/FE?
- What software licence *should* we use, then?
- What do 'open standards' and other vague terms mean?
- Who will judge whether the rules have been followed?
- Who will assist in archiving and disseminating software?
- What does 'exploited within an academic community' mean?
- Is this policy integrated with an EU directive?
- How is the JISC responding to the policy?
- How do I go about re-assessing my institution's IT policy?

Australian Government Information Management Office *A Guide to Open Source Software for Australian Government Agencies* contains good advice on:

- Degrees of engagement in open source
- Risk mitigation: the key risks involved in open source software are the same as those in proprietary software, with *additions*:
    1. There can be multiple, independent, primary vendors for open source software to choose between.
    2. As well as the usual range of support options, you can bring new feature development in-house and modify the core source.
- Licensing: includes six different scenarios and considers which of the more popular open source licences are likely to be applicable for each.

Remember that

- Saving money is not always the best starting point.
- All take and no give is not conducive to the health of open source.
- HE does often have significant technical expertise in-house.
- The academic side of HE is not used to proper software procurement

■ ■ ■

The JISC open source policy (draft)

The JISC policy was developed by OSS Watch in June 2004, revised in October 2004, and submitted to JISC committees in February 2005. It covers:

1. Policy guidelines for JISC when writing calls for proposals, ITTs etc

2. Policy guidelines for JISC services (and JISC projects generally)

3. Policy guidelines for JISC-funded software development activities specifically

- Copyright ownership of software, diagrams, schemas, documentation, manuals, user interface and source code must be recorded, and may be vested with a JISC-appointed body
- Projects must maintain an IPR register, listing all contributors to their software and who owns the copyright on contributions
- The ownership of code which is to be developed in joint projects must be established before work begins

- Copyright ownership of software, diagrams, schemas, documentation, manuals, user interface and source code must be recorded, and may be vested with a JISC-appointed body
- Projects must maintain an IPR register, listing all contributors to their software and who owns the copyright on contributions
- The ownership of code which is to be developed in joint projects must be established before work begins

- Copyright ownership of software, diagrams, schemas, documentation, manuals, user interface and source code must be recorded, and may be vested with a JISC-appointed body
- Projects must maintain an IPR register, listing all contributors to their software and who owns the copyright on contributions
- The ownership of code which is to be developed in joint projects must be established before work begins

- Copyright of software, documentation, design materials, manuals, user interface and source code must be released under an OSI-approved open source licence, unless the bid explicitly argues why this should not be the case and proposes an alternative licence.

- Software must in any case be licensed and publicly available, for any use and at no financial cost, throughout UK higher and further education

- The open source licence most appropriate in any given circumstances will depend on the mechanism chosen for exploitation and/or on-going development.

- Copyright of software, documentation, design materials, manuals, user interface and source code must be released under an OSI-approved open source licence, unless the bid explicitly argues why this should not be the case and proposes an alternative licence.

- Software must in any case be licensed and publicly available, for any use and at no financial cost, throughout UK higher and further education

- The open source licence most appropriate in any given circumstances will depend on the mechanism chosen for exploitation and/or on-going development.

- Copyright of software, documentation, design materials, manuals, user interface and source code must be released under an OSI-approved open source licence, unless the bid explicitly argues why this should not be the case and proposes an alternative licence.
- Software must in any case be licensed and publicly available, for any use and at no financial cost, throughout UK higher and further education
- The open source licence most appropriate in any given circumstances will depend on the mechanism chosen for exploitation and/or on-going development.

Trademarks Use of trademarks to establish reputation and trade on association is up to the project, not the JISC.

Patents Any patent applications associated with the project should not interfere with free distribution of software.

Dependencies Projects must record which associated software is needed to make their work run.

Archiving All documents and software code must have a preservation and archiving strategy.

Testing and quality assurance All software must have a testing framework in place, and demonstration of standards compliance.

Version control All software must be developed using
version control software, and the history must
be preserved by the project.

Sustainability and communities Projects should, where
appropriate, encourage and support user and
development communities.

Documentation Documentation must archive all forms of
documentation, including mailing lists and
forums.

Software development and maintenance Software should
follow good engineering practice, and be
demonstrable to, and testable by, peer
communities.

- - -

We need to deal with:

- individuals contributing to open source software
- staff creating software which they want to open source
- teachers making online resources
- research projects collaborating with industry
- partnerships with other academic institutions

- any act of creation generates copyright—it does not have to be claimed
- most academic contracts specify that all creations are property of the employer
- usually, there are specific exclusions for books and articles
- copyright in learning materials is usually claimed by the university
- the employee has a duty to assist the university in exploiting any created material
- software is hard (but not impossible) to patent

- the university's exploitation system for software only knows about selling licences
- the university does not have a revenue-sharing arrangement for consultancy-based exploitation
- the lawyers are reluctant to sanction open source exploitation because they see it as liability without revenue
- if the university relinquishes copyright, it is at the risk of having to buy back a later release of the product

Examples of (e-learning) open source exploitation in academia

uPortal portal framework, development by top American universities ('stone soup' group) to meet their specific needs

Bodington Small UK open source VLE, developed by Leeds, Oxford, UHI; community based on shared problems

Moodle Simple but very effective VLE, distinguished by its exemplary open source community

LAMS innovative e-learning mediating framework, new work being funded under an open source model

The primary concerns for an educational institution's IT procurement strategy should be *demand* (that is to say, why do we need the system) and *value* (what will it cost us). Beyond that, the single most important consideration is the preservation of data and the interoperability of systems.

1 *New software acquisitions should demonstrate conformance to open standards and interoperability with open systems.*

At each point on the procurement and deployment chain,
software should be assessed on its merits.

2 *Open source and proprietary software options should
be assessed using the same criteria, considering of
total cost of ownership over the expected lifetime of the
deployment.*

An institutional IPR policy should acknowledge the
significant role played by open source methodologies in
terms of potential exploitation routes.

3 *Software development by staff and students must maintain a register of IPR.*
4 *Software for which the copyright belongs to the institution must be exploited.*
5 *Open source licensing must be available as an exploitation method, and will be the default method where no alternative is proposed.*
6 *Income derived from services and training associated with an open source product must be shared with the developers.*
7 *The open source licence chosen should ensure that the institution is able to freely use all future versions of the software.*

There must be procedures in place so that staff can do work on open source projects in good conscience, without removing the protection afforded to the institution by retention of copyright.

8 *A register of offically-deployed open source software must be maintained for each unit.*

9 *A register of open source software for which staff may contribute code, documentation and support must be maintained for each unit. It must say whether contributions remain the property of the institution, or whether copyright has been assigned to a body maintaining the software.*

10 *Staff and students may deploy additional open source software for research or teaching, but may not contribute institutional intellectual property to it without explicit permission.*

- ■ ■ ■

Conclusions

These are things we want to get into IT policies in the UK:

1. Embracing open source software is an attitude, not a binary choice
2. Open source is a valid, and a good default, exploitation route
3. Standards for your data are as important as current ease of use
4. There is no stasis. Things will always change