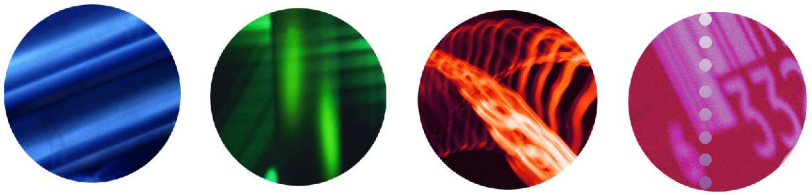
JISC

Joint Information Systems Committee

Developing JISC Open Source Policy and Practice

Alan Robiette, JISC Development Group

<a.robiette@jisc.ac.uk>



Supporting education and research

What are policies for?

- A policy is a high-level statement which aims to establish a set of practices for the general good of the organisation or community
  - Normally a top-down exercise
  - But to be successful, it needs to command a broad degree of consensus
  - Some common examples: health and safety policy, procurement policy, IT security policy ...

How do you develop one?

- Be clear about
  - What the policy sets out to achieve
  - Who its target audience is
  - How the messages are best put across
  - How responsibilities are defined and allocated
  - And (in most cases) what will happen if the policy is contravened
  - Most of all, make sure that what the policy says is practical and can be implemented

Background

- e-Government Unit Policy on OSS
  - Creates obligations for JISC (and other parties, including Research Councils and DTI)
- Top level JISC Strategy
  - References OSS (in connection with interoperability, standards and digital preservation)
- A current major preoccupation with sustainability

The main drivers

- Return on investment of public funding
  - Directly: i.e.
    - In JISC's internal service development
    - In sustaining and exploiting software arising from JISC development projects
  - Indirectly: i.e. by helping universities and colleges to invest to best advantage
- National and international co-ordination of software infrastructure
  - Hard to do without open source

Advice to institutions

- JISC doesn't tell universities and colleges what to do internally!
- But it does provide advice and support where required
  - OSS Watch: neutral advice on all aspects of open source software
  - JISCInfonet: advice on procurement and evaluation
    - Recent "compare and contrast" report on proprietary, open source and managed services; cf. Michael Coen's talk later today

JISC production services

- Example 1: JISC data centres
  - Past services have sometimes been built with proprietary components
    - Often creates lock-in
    - Tends to inhibit flexibility, e.g. if service needs to be relocated
  - JISC policy needs to address these issues
- Example 2: access management
  - Athens service was once wholly closed
    - JISC now investing in Shibboleth (open source)
    - Athens itself becoming much more open

JISC development projects

- Not always primarily directed at software development
  - Much JISC development is concerned with soft factors, proof of concept etc.
- But there has been a significant shift in recent years
  - E-learning tools & technologies work
  - Middleware programmes (including e-libraries/information environment)
  - Growing e-research involvement

What happens next?

- Historically, a mixed picture
  - In the absence of clear policy, the continuation plan has been left to projects themselves
    - Not necessarily a death warrant! Cf. The example of MailScanner (more detail shortly)
  - But arguably leaves too much to chance
- Some software products have received on-going funding
  - Usually because they have found their way into a JISC production service

MailScanner

- Funded by a £20K 1-year grant in 2000/1
  - Now "the world's most widely-used email security and anti-spam system"
  - Over 330,000 downloads
  - Over 40,000 active user sites
    - Universities, government depts, companies ...
  - Successful community generated
    - Original author (Southampton Univ) spends ~ 1 day a week maintaining and coordinating
    - Other contributors look after specified parts of the distribution

What's the secret?

- Cornered a select OSS niche
  - Commercial competition is high-cost
  - Simple clean design, pluggable, extensible
    - Began as virus detector only (initially just for Sophos, later extended to multiple anti-virus vendors)
    - Added its own spam detection (now also optionally interfaces to SpamAssassin)
    - Just added phishing detection
    - Initially written as Sendmail utility only, now also interfaces to Exim, Postfix etc.

How can JISC do better?

- Currently considering a range of options
  - From the status quo at one end
  - To a managed JISC-badged repository, with or without maintenance of selected products, at the other
  - Product selection would be a key issue in the latter case: funds will always be constrained so selection is inevitable
  - Significant current interest in the e-Science OMII model (see later slides)

Prototype to production

- Some issues here are:
  - Robustness & usability (user testing)
  - Documentation
  - Dependencies (platforms, libraries etc.)
  - Packaging (e.g. for ease of installation, ease of upgrade etc.)
- Should JISC think of giving follow-up funding expressly for this kind of "productising"?
  - And what sort of people do this best?

What is OMII?

- OMII stands for Open Middleware Infrastructure Institute
  - Funded by the e-Science Core Programme via EPSRC for 3 years
  - Began work early 2004
  - Remit is to acquire, develop and distribute robust open source Grid software for e-Research
  - A major part of OMII's effort goes into packaging, testing, quality assurance generally

Could JISC use this model?

- A bit early to tell as yet
  - OMII has a highly focused remit (lower level Grid services)
  - JISC's activities cover a much wider range of application and function
  - OMII's budget is currently £2M+ per annum – and it's clearly not enough
  - So with any kind of realistic expenditure, JISC would have to exert *very* tough product selection processes
    - How could this be achieved?

Other models???

- So far we've mainly talked about ensuring continuity
- What about the real problem: maintenance (or maintainability)?
  - Could JISC do more to create and foster self-help communities?
  - Are there ways to bring OSS service companies into the equation?
  - Could JISC broker subscription models in which customers pay for support?

The draft JISC policy

- Very much work in progress ☺
  - Much of the initial input by OSS Watch, then bounced back and forth with JISC
  - Has been seen (once) by a number of the key JISC committees
  - Their comments need to be incorporated in a further draft
  - Some changes in presentation being contemplated to improve impact
  - But – overall – not likely to undergo radical change from here onwards

So what's in it?

- In three parts
  - Policy guidelines for JISC when writing calls for proposals, ITTs etc.
  - Policy guidelines for JISC services (and JISC projects generally)
  - Policy guidelines for JISC-funded software development activities specifically
- N.B. Nothing as yet about advice to institutions
  - Should this be added? Would it improve the policy as a whole?

The content

- The first two sections are by and large uncontroversial
  - Almost motherhood and apple pie, you might say ...
  - But worth making explicit just the same
- The software development section has attracted most comment
  - Do some projects require stricter standards than others?
  - Licensing is always a sensitive issue

Summary & conclusions

- Sustainability (of services and software) is perhaps JISC's biggest problem
  - Open source policy can contribute significantly to this
  - But getting the model right is crucial
- The real achievement would be to foster viable communities without overmuch subsidy from JISC
  - And the big problem here is selection

Discussion ...