Affirmative Argument

Paul Jones
Director of ibiblio.org; Clinical Associate Professor, School of Journalism and Mass Communication and School of Information & Library Science, University of North Carolina at Chapel Hill Email: pjones<at>metalab.unc.edu

Open source [1] developers and users are unusually passionate about their work, unusual in ways that make things work well. So let me begin passionately as we talk about open source as the solution for support of institutional repositories.

You want to use, you must use, open source software for your institutional archives. Any other choice would be un-archival and unsustainable in the long run.

Now that we have that behind us, let’s discuss some of the myths and some of the reasons for dedicating your institutional repository to the use of open source software, open standards and open formats which, I contend, are inseparable.

The problem in engaging in the argument over whether institutional repositories should always use open source software is that the negative side will constantly chase the odd case that may not fit the general rule in hope that if they can accumulate enough specific odd cases falsification will seem to occur. This is Karl Popper’s falsifiability from his landmark,

Negative Argument

Michael Day
Research Officer UKOLN, University of Bath Email: lismd<at>ukoln.ac.uk

A quick glance at the most recent statistics produced by the OpenDOAR Directory of Open Access Repositories suggests that the vast majority of existing institutional repositories are currently built upon open source software. For example, the tables show that at the end of January 2009, almost half (47 percent) of the repositories listed in the directory use one of the two leading open source repository packages [8]. While this prevalence demonstrates that there is certainly a market for open source repository software, it does not necessarily follow that all repositories should be built upon it. To argue this point is not to suggest that there is anything fundamentally wrong with the open source development model itself. The open source philosophy has proved itself to be a very successful model for software development. It has also been a major inspiration for the collaborative models that underpin many recent Internet developments as well as for the concept of open science [9]. In the institutional repository context, however, there are a number of reasons why an insistence on open source software solutions may not be strictly necessary.

The first reason relates to the ever-changing technical context of repositories. Clifford Lynch’s definition of institutional repositories emphasizes that they are not “simply a fixed set of software and hardware” [10]. While at any given time repositories will have to be supported by a set of technologies, Lynch argues that they essentially constitute an organizational commitment to the ongoing stewardship of the digital content created by
Logic of Scientific Discovery [2]. But Popper’s empirical falsification approach was challenged if not overturned by Thomas Kuhn’s notion of the paradigm shift, which he detailed in *The Structure of Scientific Revolutions* [3]. I cite these as a warning not to miss the shift in software practice while being caught up in falsification’s web.

Institutional repositories have taken a few knocks in the six years since Cliff Lynch’s “Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age” appeared in *ARL 226* [4]. But I’m concerned more here about the upcoming crashes than the bumps we hit on the road to more settled standardizations.

While we have had a number of attempts at choosing standards to aid curation of the materials within our repositories, we are and will be hostage to changes in formats across time. Curation will always include migration as Lynch notes in his article. Migration in turn requires an understanding of the original formats as well as a consideration of the target formats. Formats are often bound to the context in which they are stored and retrieved. Thus access to the code that controls the items and their formats is almost as important as the formats themselves. Thinking ahead to allow for maximum ease (or least pain) for the inevitable migration of formats that curation entails means planning for long-term access to the code that controls the repository environment. Some vendors of various software solutions offer to put their code into escrow for future use were they to go out of business; however, this provision is no substitution for full open and continuous access to the code.

Proprietary software vendors often try to finesse the open source access promise by offering small customizable ports of entry into their code, usually as application program interfaces or APIs. Like software escrow promises, this is a short-term
solution to our long-term problems in curation of our valuable materials within our repositories. APIs do reduce the workload on an individual programmer, which is why many open source solutions also offer APIs. APIs are programming best practices at the moment and are not a viable alternative answer to access to the code itself. At best, APIs in proprietary software offer a temporary and brief – particularly in the lifespan of a repository – opportunity for interoperability with other systems. If a certain service is needed and if only proprietary software with APIs provides that service, then a proprietary solution might be considered to bridge that gap in service. But even then, a plan should be put in place to develop and migrate to an open solution.

For a long time, it has been argued that the market, as represented by proprietary software solutions, is more responsive to the needs of users, to new requirements and to innovations. Open source is now seen as a diverse infrastructure of solutions each in competition while also free to borrow from each other. The large number of Linux distributions most obviously testifies to this diversity. The core pieces of work are borrowed, remixed, reincorporated and revised into new specialized versions of the Linux operating system making it one of the most innovative software ecosystems in the world. It is no mistake that Stephen Weber’s *The Success of Open Source Software* [5] was 2004 winner of the Professional/Scholarly Publishing Annual Award Competition, Computer and Information Science. Weber explains how competition and cooperation flourish in open source by driving a market for innovation that is closely tied to customer satisfaction and customer participation rather than customer lock-in.

Indeed the strong market presence of open source has even converted some of its harshest critics. Recently Sun Microsystems leader Scott McNealy was quoted [6] as saying, A third set of reasons why open source software should perhaps not be viewed as the only acceptable approach to institutional repository development relates to the nature of the open source process itself. Open source software, by its very nature, tends to be developer driven. In itself, this attribute need not be a problem. However, in the repository domain, this can result in a mismatch between specific institutional requirements and what software is actually able to provide at a given moment of time. While in an ideal open source context, collaborative community development would be able to fill gaps and resolve many of the other potential conflicts, the anecdotal examples provided by Dorothea Salo in her recent article on institutional repositories suggests that the current situation is far from perfect. While recognizing many of their benefits, she comments that the three main open source software offerings currently “offer varying quantities of installation and maintenance headaches, expensive hardware demands, customization and development hassles, and poor fit with existing library software, websites, and services” [12]. Similarly, a 2007 report for UNESCO’s Memory of the World program suggested that one of the major open source repository platforms “has evolved into a monolithic software application, and complex code base, that introduces potential scaling and capacity constraints for some large institutional users” [13]. While it is fair to point out that these problems are certainly not unique to open source, it may be a signal that software development in the repository domain is currently immature. Certainly the rapid development cycles typical of open source software can make the local customization of repositories problematic. Time spent carefully redesigning repository interfaces to meet local needs can be wasted when updated versions of repository platforms are released. Solutions might include the modularization of repository platforms combined with the promulgation of consistent and stable standards and APIs.

In her article, Salo suggests that some repository software platforms need to be more responsive to specific institutional requirements, for example with regard to things like mediated deposit or the batch import of documents. There may also be a need for repositories to interact more closely with a wealth of other institutional systems, which are currently typically based on a mix of proprietary and open source solutions. While a recent report suggests that institutions in the United States (at least) might welcome additional open source development within the higher education sector [14], it might seem perverse in institutional terms to insist that repositories require an open source
“If you think about it, proprietary software is the software equivalent of a planned economy led by dictators, whereas open source is all about choice, the market economy and multiple competitive players.” While McNealy is as usual bombastic, he does have a point. Open source is not monolithic by any means. If a project spirals out of hand, gets too cumbersome or simply disappoints you, there are a variety of alternatives available. In the case of institutional repositories, the Open Society Institute’s Guide to Institutional Repository Software version 3.0 [7] lists nine products serving our repositories that are Open Archive Initiative compliant; only one is proprietary. Open source particularly in the area of institutional repositories is a lively and competitive – yet cooperative – area of development and will continue to be innovative and responsive.

The ends and the means of institutional repositories are one and the same. The infrastructure that supports open access needs to be open itself. The task of curation includes the task of migration, the task of copying and in many cases the task of restoring and renewing the contents of the repository, the context of the repository and even the software infrastructure. Software, digital formats, even standards are at this time far from settled and further from being set in stone. As our conversation about what an institutional repository should be, how it should be managed and what should be included continues our software must be flexible, responsive, customizable, innovative, inclusive and un-owned – open to all to improve. Only open source solutions will and can insure our success.

Finally, and perhaps most importantly, the insistence that institutional repositories should always be built on open source software – regardless of context – would seem to be unnecessarily focused on the means rather than the ends. The purpose of any repository should be the stewardship of well-managed collections of institutional content. Therefore, any focus on openness should be concentrated on making sure that repository content and its associated metadata can be exposed to other systems through tools like the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) and that both content and metadata can be exchanged successfully between repositories and other systems. In the same way that repository technologies will change over time, these interoperability mechanisms will also need to evolve to take account of new opportunities for sharing data. The experience of some data science domains suggests that there is a need to focus a great deal of attention on adherence to open standards and on the development of stable APIs, as well as on shared approaches to semantics [15].

To conclude, where open access is the main objective of an institutional repository, the exact license status of the software that underlies it does not seem particularly significant. While the statistics from OpenDOAR suggest that most repositories are currently developed on open source platforms, a growing market for outsourced solutions exists, including for those provided by the commercial sector. In the longer term, however, things could become even more complex. For example, institutions could contract out some core repository functions to third party services based in the cloud [16]. Simultaneously, however, repositories are also likely to depend increasingly on their tighter integration within a more complex set of institutional systems and processes (for example, as part of research workflows) and in many cases linked to national and international research e-infrastructures. The open source development model is likely to have a very significant role to play in helping to develop and link these complex infrastructures, but other approaches will still remain viable.
Institutional Repositories: The Great Debate

Resources Mentioned in the Topic 3 Debate


