Open Standards Versus Open Source in E-Learning

The easy answer may not be the best answer

By **James Dalziel**

The first decade of the twenty-first century is witnessing the convergence of three strands of development in e-learning, strands that had relatively independent origins during the 1990s. The first is the development of e-learning technology as a recognized industry. This is based on the rise of the Web and the widespread adoption of e-learning software and courses, especially learning management systems (LMSs) such as WebCT and Blackboard in the education sector and Saba, Click2Learn, and others in corporate training. This strand grew predominantly from software innovation around proprietary e-learning systems and has found its way into the wider market through venture-capital investment.

The second strand arose from attempts to create open standards for e-learning software and content, driven by specification organizations such as the IMS Global Learning Consortium, Aviation Industry CBT (Computer-Based Training) Committee (AICC), and Advanced Distributed Learning (ADL) network sponsored by the U.S. Office of the Secretary of Defense, and relevant committees of international standards bodies, such as the IEEE Learning Technology Standards Committee. Despite the potential relevance of these open standards for the proprietary e-learning systems of the first strand, the consistent adoption of e-learning standards by LMS vendors was slow, particularly in the education sector.

The third strand is the much wider open-source software movement, which has produced highly successful software such as the Linux operating system and the Apache Web server. While open-source software has both historical and philosophical roots within universities, e-learning was not one of the major focus areas of the open-source software movement during the 1990s.

There have been exceptions that crossed the boundaries among the three strands. The major impact of each strand, however, tended to occur without relying on the others during the late 1990s. E-learning technology rose to fame largely without standards or open-source software; e-learning standards were initially developed without widespread vendor adoption or open-source software examples; and open-source software focused its major efforts on basic infrastructure, such as operating systems and Web servers, rather than specific applications, such as e-learning. In addition, open-source software tended to focus on rapid innovation rather than the slower, consensus-building approach that is typical of open standards.

A number of open-source software development efforts specifically target e-learning

All this is changing. The past few years have seen wider adoption of standards by e-learning vendors, especially in corporate training, through the adoption of the Shareable Content Object Reference Model (SCORM) from ADL, which builds on work from IMS and AICC. At the same time, a number of open-source software development efforts specifically target e-learning, such as the Open Knowledge Initiative (OKI) at the Massachusetts Institute of Technology and a range of Joint Information Systems Committee (JISC) projects in the United Kingdom.

Given this context, it is timely to ask some questions about the relationship between open source and open standards in e-learning software development. While it seems that open-source software that implements open standards is an attractive model for many users, two questions are becoming increasingly pressing at this stage of the e-learning industry: Which is more important—open source or open standards?, and, What problems could arise from open-source e-learning software that implements open standards? These questions are particularly relevant to the education sector and to government policy related to this sector.

Open Source or Open Standards?

Open-source software is based on open distribution of the source code that forms the software's foundations. This means that any technically competent programmer can examine the inner workings of the source code and make changes to the operation of the software. Open-source software is typically provided free of charge or for a nominal distribution cost. Some open-source licenses require that any changes to the source code be redistributed on the same open-source license terms as the original source code.

Open standards are transparent descriptions of data and behavior that form the basis of interoperability. Interoperability is the ability of different software systems to exchange information such that the systems can act in equivalent ways on the information, resulting in equivalent user outcomes. In practice, interoperability means that users are not locked to any one software system—they can substitute one standards-compliant system for another. Open standards can be implemented by commercial systems and open-source systems alike. Provided that all systems adhere to the same standards, there is no impediment to combining commercial and open-source software systems.

It may be natural, then, to consider open source preferable to open standards. This is because in open-source software development, all of the source code is freely available, and, if it does not correspond to open standards, it could be modified to be standards-compliant. Commercial systems that support open standards rarely provide access to their source code, so external developers cannot change the software as desired. Hence, were we forced to choose, open source would appear to be the more flexible option.

In practice, it might not be so simple. The open-source code may be sufficiently complex that a high level of technical expertise is needed to modify the code. As a result, the cost of doing so (in time and/or money) may be considerable if the modifications are substantial. In addition, the changes may not be compatible with the intention of the original software, which can cause a "fork" in development that splits the original open-source developer community into separate groups, potentially weakening both efforts.

Most open-source e-learning projects have not arisen spontaneously from the goodwill of freelance software developers

Most open-source e-learning projects have not arisen spontaneously from the goodwill of freelance software developers. They are typically the result of government or foundation funding, where developers are paid for their contributions to the project (either as contractors or as salaried employees of organizations such as universities). In the wider open-source movement, a voluntary community of developers supports projects such as Apache and Linux, hence their ongoing development is independent of the vagaries of project funding. This is not the case in e-learning, making any given open-source developer community highly susceptible to collapse when project funding ends. This is a major problem, not well understood by governments or foundations that currently provide funding. The problem arises from the difference between "traditional" open-source developer communities (which tend to be self-sustaining) and funded-project developer communities (which often evaporate when the money dries up).

This is not to imply that there can be no useful outcomes from government- or foundation-funded e-learning open-source software projects. Even when the project ends and the community of programmers disbands, the source code may be useful to others if it is made widely available and, in particular, if it is well documented (although poor documentation of open-source software is endemic). This piecemeal, ephemeral approach to open-source e-learning software development has many downsides, however, such as loss of specialist expertise (when hired programmers move on to other areas as funding ends), loss of continuity, risk of duplication (where ephemeral projects are not widely known), and, perhaps most importantly, a lack of interoperability (where standards are not implemented).

Lack of interoperability in open-source e-learning development can be illustrated by the new IMS Digital Repositories Interoperability (DRI) specification and two recent e-learning initiatives. The OKI project is based on a range of open source e-learning service APIs, including a digital repositories service. OKI has collaborated closely with IMS over the past year—the same period in which the DRI was developed. At the February 2003 Vancouver IMS meeting, however, when questions were asked about interoperability between the OKI digital repositories service and the new IMS DRI specification, it became clear that the two were not compatible. It is unclear why the OKI project, which has had significant funding, high publicity, and a close association with IMS, should choose to implement its own proprietary digital repositories approach, despite the open DRI specification that was developed "next door." Cambridge University's e-learning research center (CARET) recently tried to implement OKI and IMS DRI approaches together and acknowledged that this was more difficult than they had hoped.

The second initiative is the Australian K–12 Le@rning Federation's open-source development of a Basic E-Learning Toolkit (BELTS). A component of BELTS is a messaging service for transportation of learning objects among BELTS servers. Messaging services among repositories are described in the DRI specification, but BELTS does not yet implement the DRI specification.

Lest there be any doubt about the DRI specification's usefulness, it has been successfully implemented by at least two major projects, the Learning Object Network's collaboration with ADL and the Australian Collaborative Online Learning and Information Services (COLIS) Demonstrator project. Both projects have received international acclaim at IMS and elsewhere. In each case, the open DRI specification has proven to be invaluable in solving difficult digital-repository problems, providing a foundation for interoperability with both open-source and commercial products that adhere to the DRI specification. Neither OKI nor BELTS currently supports interoperability other than by using their own proprietary (but open-source) interfaces.

The point of these examples is that open source is no guarantee of interoperability, while commercial (and open-source) systems that implement accepted open standards provide a strong foundation of interoperability. The success of SCORM in the United States, particularly for corporate training, indicates that commercial e-learning vendors can benefit from interoperability to the point where a standard becomes a de facto requirement and nonconforming systems risk commercial ruin. In addition, the widespread adoption of SCORM has fostered rapid development of a large, highly professional e-learning courseware industry for corporate training (for example, SkillSoft, NETg, DigitalThink, and others), which might have been immature and highly fragmented in the absence of a unifying standard to foster interoperability.

So, which is more important for e-learning today—open source or open standards? In my view, open standards are more important for two reasons: first, interoperability is key to any distributed e-learning systems environment, and second, we lack a robust open-source developer community to adapt project work that is not standards-compliant.

Open Source That Implements Open Standards?

Given the discussion above, creating open-source e-learning software that implements open standards might appear to be an attractive option. The resulting software would have open-source code that is available for inspection and adaptation while conforming to relevant open standards, ensuring interoperability.

Open source is no guarantee of interoperability, while commercial (and open-source) systems that implement accepted open standards provide a strong foundation of interoperability

One of the most important potential problems with open-source software, however, even when it implements open standards, is the lack of support typically expected from commercially developed software. Whom do you call when you have a problem? There are no help lines, no on-site visits, and, in the absence of a strong open-source developer community with sufficient goodwill to support users, there may be no readily available support at all (for example, try getting support for the IMS Content Packaging tool, Pacman). This lack of support is true for most open-source e-learning software today and for the foreseeable future. As a result, it is usually necessary for an organization to develop internal expertise in open-source systems, which itself is risky if this expertise resides in a single person who could leave the organization at any time.

While commercial software and open standards are often perceived to be in conflict, this is not necessarily the case. Indeed, it may make good commercial sense to use open standards where a product needs to work in harmony with others, even though the internal workings of the product are proprietary (rather than open source). In some cases, interoperability arising from license maintenance for standards-compliant commercial software may be more cost effective than “free” (open-source) software that may have high implementation, integration, and support costs (in terms of staff time). The problem is that many educational organizations do not account for staff time and external license spending in equivalent ways, leading to potentially false economies in choosing “free” software, which often has significant (but uncosted) staff support requirements.

In the current context, a commercial system that implements open standards has an important advantage over limited-life open-source projects. A commercially successful product usually ensures ongoing support, maintenance, and future development arising from product revenues. This may provide a more stable and ultimately cost-effective mode of support than relying on a small (or nonexistent) open-source developer community that relies on project funding.

In other words, open-source, standards-conformant e-learning software is not free of risks and is not necessarily the most cost-effective option where commercial vendors have implemented open standards and demonstrated easy interoperability.

Where to Next?

The points made above should not be seen as a final word but a contribution to the debate about the direction of open source and open standards in e-learning, and governments and foundations need not stand on only one side of this debate. It is worth investigating and understanding the differences between open source and open standards in software development and systems integration. We have much to learn about the strengths and weaknesses of open-source software as it moves into the mainstream of software development and systems integration, be it for e-learning or other areas. Equally, the theoretical case for standards is easily made, but difficulties of implementing standards require further investigation. Even open-source, standards-compliant software is not without practical problems when it comes to implementation.

In terms of open source, if government or foundations want to address the problems identified above, they will need to find ways to assist both the open-source developer community (to ensure its viability and continuity) as well as the user communities who implement and use these systems (to ensure open-source support costs do not become prohibitive and too risky). Efforts to address problems with open standards will need to find ways to assist the development and implementation of standards, particularly through demonstrator environments that test the practicalities of implementing open standards. In addition, the formation of standards testing/certification organizations (monitored by government/ foundations) can encourage wider adoption of standards through assurance of compliance with those standards. The creation of "application profiles" based on specific local or sector-specific needs can further facilitate the successful uptake of open standards.

While commercial software and open standards are often perceived to be in conflict, this is not necessarily the case

In either case, it appears that the initial investment by government and foundations to support open-source and open-standards efforts needs to be complemented by further investment to provide continuity and critical mass within the communities they seek to foster. This will ensure that potential benefits are not lost due to the evaporation of funding over the short term. A longer term view (5–10 years and beyond) is needed to realize the maximum benefits of investments in either area, and this will require a new concept of project funding in many instances because the life span of open-source or open-standards efforts is rarely guaranteed beyond a few years. I believe this is the most important current challenge for both areas.

Finally, we may see the rise of a new kind of software company, one designed to overcome the obstacles identified above. This company would develop and support open-source software based on open standards. It would provide long-term support and software development contracts to organizations that want to adopt open-source, open-standards systems but are concerned by the risks identified above. By aggregating open-source development and support contracts, a viable commercial enterprise may arise, though based on a radically different business model from that of traditional commercial software development. It remains unclear as to whether this outcome will be preferable to the traditional commercial software development process. In either case, the value of open standards remains. *e*

James Dalziel (james@melcoe.mq.edu.au) is Director, Macquarie University E-Learning Centre of Excellence (MELCOE).