Creating a Collaborative Information Technology Environment for Higher Education

Ira H. Fuchs

Wired to Cooperate

The *New York Times* recently reported that researchers at Emory University in Atlanta used magnetic resonance imaging to scan the brains of a subject population participating in a simulation of the classic Prisoner's Dilemma.[1] They expected that subjects who failed to maximize their gain by cooperating with one another would exhibit the greatest brain activity. To their surprise, the strongest signals arose as a consequence of cooperative alliances and in areas of the brain known to respond to positive stimuli such as desserts and pictures of pretty faces. These findings suggest that humans are predisposed to cooperate. Why, then, is the degree of information technology collaboration within higher education not yet what it could or should be?

Higher education has not collaborated well in the past for a number of reasons, but I believe we are now on the brink of a new era. In particular, middleware, a layer of software that mediates between applications and the infrastructures on which they run, offers an approach that may help us produce and share high-quality courseware and construct administrative applications more cost-effectively and more efficiently. Such new collaborative projects will satisfy higher education's unique functional needs, integrate with our existing campus infrastructures, and be relatively easy to support.

The hope and expectation is that a robust middleware layer will make it easier for campus and commercial developers to produce software that will work in higher education's highly heterogeneous environments and demonstrate our ability to share information and solve our problems collaboratively. Two projects, partially funded by The Andrew W. Mellon Foundation, offer great promise: the Open Knowledge Initiative (OKI), a major middleware initiative, and OpenCourseWare (OCW), a stunning example of how an institution can share its course materials with faculty and students throughout the world.

Looking Back to NeXT

At the Educom annual conference in Long Beach, California, in 1987, Steve Jobs and H. Ross Perot unveiled the NeXT cube—a striking, black workstation aimed squarely at the higher education market. Based on a 33-MHz Motorola 68030 processor that was better and faster than the microprocessors found in IBM PCs or their clones at that time, the box contained optical technology rather than conventional magnetic hard disks. It came bundled with word processing software, Mathematica, a dictionary, and the complete works of Shakespeare. Its user interface was an innovative model of simplicity. Badgered about the cube's long-overdue launch, Jobs replied, "Late? This computer is five years ahead of its time."

More breathtaking than the cube itself was its application development environment. Jobs's team used fledgling object technology to build an environment that permitted users to point-and-click their way to the creation of their own applications. NeXT proposed that institutions of higher education jointly assemble an object library that would run on any computer running UNIX and a layer of software provided by NeXT. Jobs's vision: to encourage higher education to create shareable software modules. We would all share objects and applications in much the same way that we share journal articles and monographs.

Three years later, in virtuoso style at the London Palladium, Jobs demonstrated how simply one could create and use applications on his workstation. Around that time, Tim Berners-Lee decided to use the NeXT to develop his idea for linking academic institutions across the Internet. The World Wide Web was spun on the NeXT because, as Berners-Lee wrote in his book, *Weaving the Web*, "The NeXT interface was beautiful, smooth, and consistent. It had great flexibility and other features that would not be seen on PCs till later."[2]

At twice the price of an IBM PC or Macintosh, the $6,000

NeXT cube failed because it was impossible to get most institutions of higher education to buy into a single platform or development environment, particularly given the nature of the competitive marketplace. Berners-Lee concluded, "NeXT required users to accept all these innovations at once—too much." It was a computer system that higher education simply refused to embrace. The system came with a considerable amount of academic software, but there were too few developers, and higher education produced very few new applications. Jobs sold only 50,000 units. NeXT confronted the classic dilemma facing any new computing platform: Buyers want to see applications before they buy the hardware, and developers want to create applications for a large established base. Jobs had worked his magic at Apple in attracting software developers to the Macintosh, but he failed to get the developers excited about the NeXT. As my colleagues lamented, the only problem with the software that came with the NeXT was that there wasn't enough of it.

Jobs's NeXT cube, though a wonderful machine in many ways, never realized his dream. But his vision of a distributed object framework, a sharable object library across a network of compatible machines, was right on target.

The Importance of Collaboration

Colleges and universities compete for students and faculty, and they certainly compete on the playing field. However, unlike other "industries," higher education has a culture and an ethos of sharing the fruits of much of its labors—whether it be published research, textbooks, course syllabi, or locally developed software. This willingness to pool resources for the common good is higher education's unique advantage, and it offers the key to the cost-effective use of information technology.

Over the course of the past two decades, teaching and learning have been invigorated by multimedia presentations, simulations, online tutorials, interactive applications, and access to knowledge databases. More and more faculty are using presentation graphics and requiring the use of specialized hardware and software resources. Many faculty would like to further integrate technology into their research and teaching and move from being the "sage on the stage" to the "guide at the side." These transformations often involve a dramatic increase in workload associated with the preparation of multimedia courseware and the maintenance of technology-enhanced courses. Collaboration on middleware can reduce the efforts required to create and maintain educational software.

Administrative applications are another example of higher education missing the opportunity to share solutions to common problems. Colleges and universities have recently invested huge amounts of time and money to replace aging "homegrown" administrative applications (human resources, financial systems, student systems, inventory, fund raising, and so on). The initial goals were laudable. New, distributed, Web-accessible applications sought to take advantage of workstation user interfaces and the power inherent in a distributed architecture. New systems, the thinking went, would deploy a standard, modern information technology infrastructure; improve functionality for central and departmental users; adapt quickly to changing conditions and to demands for information; simplify the storing, searching, and reporting of information; and move systems off expensive centralized mainframes.

Instead, most academic institutions have struggled with their implementations. Many campuses are concluding that higher education cannot readily adapt to systems primarily designed for for-profit organizations. As a result of the compromises inherent in using these applications, campuses are finding that they often have reduced rather than enhanced functionality, and that the systems are relatively inflexible, requiring institutions to spend additional large sums to make significant changes to their business rules. Many campuses are concluding that the new applications have substantially increased the cost of system ownership. Some have concerns about the long-term viability of system vendors within the higher education marketplace.

The aggregate cost of making these changes, academic and administrative, across all of higher education is enormous. Schools such as Stanford, the Massachusetts Institute of Technology (MIT), and Princeton have already spent upwards of $100 million each to replace a mere subset of their administrative systems.

Hence, the most cogent reasons for collaboration: cost and time savings. Even many relatively wealthy schools are finding that "going it alone" is no longer financially feasible. The University of Michigan, for example, recently decided to cancel a project to develop portal software for academic and administrative departments after working on it for one year and spending more than $1 million. Officials there realized that they could take advantage of other multi-institutional efforts to create portal software, including the Mellon Foundation–supported uPortal project.

A generic approach to the development of courseware and administrative applications may satisfy most institutions and result in significant cost and time savings. Collaboration on courseware and instructional applications will have the additional benefit of tapping the creativity and ingenuity of faculty and staff at many institutions. By collaboratively developing courseware, we might simultaneously lower the cumulative costs of development and result in a far richer variety of well-honed, relevant materials from the higher education community.

Flaws in IT Collaboration

There are many obstacles to improving information technology collaboration among institutions of higher education: our highly varied campus infrastructures, the complexity associated with creating sharable software, intellectual property issues, the lack of faculty incentives to create material that can be used by others, and the difficulty in customizing borrowed material, to name just a few. Each of these impediments—as well as how the situation might be improving—is described below.

Heterogeneous Technology Infrastructures

There are many ways for a campus to validate its users, store data, access network resources, manage data archives, maintain databases, and so forth. It is almost certainly true that no two colleges or universities have completely compatible information technology infrastructures. Therefore, an application written on one campus is unlikely to run on another campus without first undergoing significant and costly modifications.

Before the dawn of personal computers and before the birth of the multitude of technology providers that exist today, campus information technology revolved around the IBM mainframe and its allied software. Designing and writing applications that could be shared among these IBM shops was relatively simple: follow the IBM standards, and compatibility was (mostly) assured. However, those halcyon days are gone, never to return. Higher education now has many more ways to meet its information technology needs, but increased choice brings with it considerable complexity and difficulty in sharing campus-developed software. Middleware offers the means to insulate developers from the technical details of the environments in which their code will run.

Local, Expensive Application Development

The development and maintenance of sophisticated course materials and instructional software has proven to be extremely difficult, time-consuming, and costly. Institutions face initial development or acquisition costs, as well as support and maintenance costs—including upgrades to applications and the costs of modifications to courseware that often fails to function whenever the base infrastructure changes. These difficulties of supporting and customizing off-the-shelf solutions to fit local computing environments are particularly daunting for smaller institutions with limited resources.

Generalized systems are yet more complex because approaches must take into consideration varied campus infrastructures and business processes. Campuses have different networking and operating system infrastructures, and unique business rules and requirements that are not easily met by standardized approaches. Far more resources are required to cover all potentialities. Higher education is not in the business of creating generalized solutions to be shared by multiple institutions, so why should an individual campus, already burdened by the high cost of creating and maintaining systems for itself, take on development costs that will have little or no local benefit? As a result of all of these difficulties, institutions have tended to focus on creating systems and courseware that satisfy their narrow, local requirements rather than seeking agreements with other institutions that would allow them to build on standards and a common set of features.

However, taking a modular approach and using middleware, institutions soon might be able to combine parts supplied by other colleges and universities or commercial providers, together with discrete parts built locally, and construct an application in less time and at far lower cost than would otherwise be possible.

Intellectual Property Rights

Intellectual property issues have become major obstacles to the creation and sharing of instructional materials. Higher education is in the unique position of being a producer, distributor, and consumer of intellectual property. Our faculty write scholarly papers and textbooks, they design and implement software as part of their teaching and research, and they create materials used in their classes. Our institutions also act as distributors when they make materials available in many forms, including over the World Wide Web. And, of course, we are major consumers of intellectual property when we buy or license materials for our libraries, laboratories, and other educational venues. At times it seems almost impossible to adopt consistent intellectual property policies that address all of these roles.

Many college courses today use copyrighted materials, such as portions of published papers, snippets of commercial videos, musical performances, and so forth. The process of preparing these materials for use inside an institution, not to mention for sharing them outside, is often dominated by the need to secure permissions from content owners. Given the fact that today's World Wide Web turns everyone into a potential publisher, identifying the content owners, contacting them, and securing permissions can be a daunting task. As part of its OpenCourseWare project, MIT found that the single most difficult problem in making course materials available on the Web was clearing the copyrights for any materials not originally created by their faculty. Some of their courses require tens and even hundreds of items to be cleared for use. As the sharing of instructional materials proliferates, there may be a place for a new organization especially attuned to the needs of higher education that would help colleges and universities secure the necessary rights to use materials both on and off their campuses.

It is worth noting that copyright laws were originally established on behalf of the public good to encourage the creation and dissemination of information and to foster progress in the sciences and the arts. For almost 200 years, the laws proved to be flexible enough to accommodate new technologies and to balance the interests of creators, publishers, manufacturers, and the user public. Over the past two decades, however, the accelerating pace of technological development has challenged the flexibility of the law to the point that existing policies appear to be discouraging rather than encouraging the dissemination of information.

The 1998 enactment of the Digital Millennium Copyright Act (DMCA) represents the most comprehensive reform of United States copyright law in a generation. Unfortunately, the act contains provisions that are particularly unfriendly to the notion of "fair use" in a digital environment. For example, as the Electronic Frontier Foundation contends, by banning all acts of circumvention—and all technologies and tools that can be used for circumvention—Section 1201 of the DMCA grants copyright owners the power to unilaterally eliminate the public's fair use rights.[3] The act is of continuing concern to those interested in the free-flowing integration of multimedia into virtual classrooms. The motion picture and recording industry alone has legions of lawyers prepared to descend upon any university or other educational nonprofit that they find using their materials without license, for any purpose.

There may be no panacea in the standing copyright and intellectual property law, but the broader legal framework has improved recently. The Technology, Education, and Copyright Harmonization (TEACH) Act of 2001, which became law in late 2002, makes it easier for educational institutions to integrate films and other audio-visuals in instruction. The TEACH Act permits institutions of learning to integrate copyrighted materials into instruction without securing the permission of the copyright holder. Under the act, instructors at accredited, nonprofit institutions may show selected portions of movies and other dramatic works. Distance-education providers are permitted to share literary and musical works. In addition, many institutions of higher education are clarifying the ownership and usage rights of the institution and their faculty for course materials. For example, as part of their recent courseware initiative, MIT has made clear that any courseware created with substantial university resources is jointly owned by the faculty member and the university and that such educational products will be shared freely for all noncommercial uses.

Projects such as JSTOR have shown that it is possible to achieve enhanced cooperation and involvement with various intellectual property stakeholders. For example, JSTOR's agreements with scholarly publishers include an updating provision referred to as a "moving wall"—a fixed period of time ranging, in most cases, from two to five years—that defines the gap between the most recently published issue and the date of the most recent issues available in JSTOR. The moving wall helped establish JSTOR as the trusted archive for journal backfiles, while also protecting publishers from losing subscription revenue. Many publishers discovered rather quickly that the availability of the electronic versions did not lessen the demand for the printed copies; indeed, in some instances, demand for the printed material actually increased. As a result, some publishers are shortening the moving wall or eliminating it altogether.

Inadequate Faculty Incentives

While many faculty believe that the innovative design and development of courseware deserve recognition and reward on par with other scholarly pursuits, decisions about advancement at most institutions continue to depend overwhelmingly on research output. Many universities, especially research institutions and top-tier liberal arts colleges, consider courseware/software development to be a less scholarly pursuit than research. At the moment, not only is courseware development not considered for tenure and promotion, it requires precious time that would otherwise be devoted to research and other activities valued in tenure decisions.

Courseware development will be helped enormously by finding ways to reward faculty who invest time in it. Faculty naturally put the greatest value on tenure, promotion, and monetary rewards, but there are other motivators. Many schools are experimenting with special "faculty incentive programs," including sabbaticals, graduate student assistance, summer stipends, reduced teaching loads, technology grants, awards, and other forms of recognition.

Liberal arts colleges are ahead of research universities in considering the development of courseware as valid criteria for faculty advancement. However, even at institutions inclined to include courseware in their deliberations, the problem of how to vet the value of these creations to higher education remains. It is relatively easy to read a research paper, check whether it was accepted by a prestigious journal, or count the number of times the paper is cited by others. It is not nearly as straightforward to determine, for example, whether the computer simulation written by a faculty member for her economics course is a valuable contribution to the teaching of economics broadly.

EDUCAUSE has attempted to address this need by distributing awards for exemplary courseware. Unfortunately, this courseware is not distributed widely, and it is frequently incompatible with other campus infrastructures. Since few schools can use the software, there is no meaningful way to judge its value in other educational settings. By eliminating the complication of the myriad campus operating environments, middleware offers a way to enlarge the potential "market" for such courseware. Through the addition of a middleware-mediated tracking service, it will be far simpler to evaluate the value of the faculty efforts.

Teaching As a Customized, Personal Endeavor

Using courseware developed by another faculty member has always been a dilemma owing to the "personal" nature of teaching. It is difficult enough to use another faculty member's textbook, but using PowerPoint or other electronic materials can be even more unappealing to faculty because the materials, which were never intended for sharing, are tailored to the style and methods of the person who wrote them. Such sentiments are understandable. Faculty might be more willing to incorporate small segments of another's lessons and lectures, but how do they locate the needed portions?

Until now, faculty have not had the tools to locate and integrate portions of lessons and educational materials developed by other faculty. However, recent efforts to make courseware more granular and to develop instructional content with descriptive, standardized metadata will permit faculty to find and incorporate small course units of relevance to their interests. Using another's course materials will no longer be an all-or-nothing decision. Faculty will be able to use the smaller parts in new contexts and in ways that will encourage and enhance personalization. As we will see, middleware provides the means for this course material to be used on many more campuses than is possible today. The more it becomes possible to share content, the more faculty will incorporate it to personalize and, in turn, share their own course materials. The easier it is to do, the fewer incentives will be required to encourage it.

Ubiquitous Network and Internet2

Over the past two decades, our investments in information technology have generated significant changes in our infrastructures. Campus computing devices have become inexpensive and ubiquitous. An order of magnitude increase in network speed has substantially eased the dissemination of information. High-speed Internet connectivity reaches most dormitories, classrooms, laboratories, and study areas. A very-high-speed Internet2 now provides our research institutions with as yet substantially unused bandwidth. As a result, we now have the ability to collaborate and to disseminate materials in ways not possible before.

Middleware

Notwithstanding all the barriers, there is hope, as I have indicated. New tools and new approaches to software development are making the creation of courseware easier. By sharing software objects and modules, and by working through collaborative development environments, we can substantially lower the cumulative threshold of effort and therefore the costs required to develop and to maintain instructional and administrative technologies. The key is middleware.

Although there are probably as many definitions of middleware as there are middleware developers, most agree that middleware is the layer of software that stands between the campus infrastructure (our networks and data repositories, for example) and the applications that run and take advantage of that infrastructure. The middleware layer acts like the middleman in a transaction to eliminate potentially dangerous, direct communication between the applications and the infrastructure and to provide safe translations between the layers. Appropriately maintained, the middleware software layer will reliably translate requests and pass along information and instructions that the physical infrastructure and individual applications will understand. As a result of this architecture, an upgrade or minor alteration in the network infrastructure or in an application will no longer cause entire systems to crash.

Examples of basic middleware services include

- authentication (ensuring that a user is who the user claims to be),
- authorization (ensuring that a user has permission to use a service or specific operation),
- management of user profiles,
- database access,
- directory services,

- content storage and retrieval,
- naming conventions,
- file sharing,
- file access, and
- access to campus external network connections.

The primary feature of middleware is the application programming interface, or API, which precisely defines particular services. Software developers can use these bundles of functionality without having to know how they are actually implemented and can be confident that they will work within the local infrastructure. The challenge is to implement APIs that successfully work both with our varied campus infrastructures and the applications that run on that infrastructure. An API for file service, for example, would need to work with the different file service infrastructures that exist on our campuses. An authentication API would need to work with Kerberos[4] and Lightweight Directory Access Protocol (LDAP), the two primary authentication infrastructures. To succeed, the APIs need to support all common campus infrastructures.

Applications that support such APIs will run within their respective institutional infrastructures. To authenticate or to use campus file services or the like, programmers would simply need to write applications that refer to these APIs. Middleware will evolve constantly to accommodate developers and advances and modifications in local infrastructures. But by modifying the APIs to reflect such technological evolution, the actual applications that call the APIs will continue to function while being buffered from the changing network and operating system infrastructure and from simple or complex changes to other unrelated applications. More simply put, educational software developers will not have to face expensive, time-consuming rewrites of application code every time an institution alters its infrastructure. And it will be possible to upgrade, maintain, and support each functional area and to add new functions without requiring costly enhancements to the entire base infrastructure.

To achieve this array of modular interoperability—to make this framework succeed where the NeXT cube failed—higher education will need a critical mass of institutions wedded to this approach and campus applications that rely upon such APIs. Constructed collaboratively, middleware holds the promise of creating courseware and administrative applications far more cost-effectively. By identifying and providing these basic services—not for a single campus or as piecemeal solutions for individual problems, but across as many real campus environments as possible—higher education will gain consistent, high-quality middleware services at a substantially lower cumulative software development cost. Duplicative development will be minimized, and services will be more uniform and scalable and will provide consistently high performance. Support services and maintenance will be simplified. Further, faculty will be much more likely to produce courseware if the tools available vastly reduce the effort required to do so.

Two Middleware Initiatives

For two decades, initiatives within higher education have focused upon networking and computing hardware, not the development of middleware. Several current initiatives are attempting to systematically address the need for middleware.

NSF Middleware Initiative

The National Science Foundation Middleware Initiative[5] (NMI) brings many institutions of higher education, the private sector, and other government agencies together to design, develop, and deploy a set of robust middleware functions and services in support of scientific research. NMI project goals include

1. Helping researchers use and share distributed computers and large, distributed data collections and remote instruments.
2. Developing advanced collaboration and communications services in support of research and education.
3. Assembling a working architectural framework that will permit Internet users to avail themselves of such services consistently and transparently.

Following thorough testing, debugging, and use, NMI released several software packages in early 2002. The early releases include an open-source toolkit that coordinates highthroughput resource sharing among large collections of multi-institutional distributed computers, and a distributed system that monitors and forecasts network performance.

Open Knowledge Initiative

The Andrew W. Mellon Foundation has funded the first two years of what will be an ongoing effort to design and develop an "architecture" for course management systems that is open-source, scalable, sustainable, and extensible. This project, the Open Knowledge Initiative (OKI), will facilitate the development and delivery of learning management systems and educational software applications. To establish a clean boundary between the various components that support educational applications, MIT and a group of collaborating, partner institutions[6] are taking a structured approach to this architectural specification.

OKI's architectural framework for APIs precisely defines how the components of a learning technology environment communicate with each other and with campus infrastructures. By clearly defining points of interoperability, the architecture allows the components of a complex learning environment to be developed and updated independently of each other.[7]

As a core component of information technology infrastructures, learning technologies must be robust, scale to meet the increasing demands from faculty and students, adapt to ever-changing technologies over time, and work seamlessly with existing campus infrastructures—as varied as those are. Above all, learning technologies must successfully support faculty and students. They must be flexible enough to adapt to a wide range of instructional requirements and styles, yet stable enough to enable faculty and students to concentrate on teaching and learning and not on the technology itself.

OKI aims to provide an architectural framework that will help learning environments meet these demands through the integration of three general categories of software:

1. Learning applications such as quiz authoring, course management, and collaboration tools.
2. Central administration tools such as student information, human resources, and directory management.
3. Academic systems such as library information systems and digital repositories of research and educational materials.

Once the architecture is fully adopted by the education market, new components could be plugged into the learning infrastructure using OKI's standardized APIs. So long as all parties conform to the architecture, colleges and universities would be able to exploit new technologies and new learning management components as they become available from other faculty or from educational vendors. Such components could be updated individually without destabilizing other components or the physical infrastructure.

Existing software and hardware vendors are not motivated to devise vendor-neutral frameworks, preferring to lock-in markets by selling closed solutions that tie their customers to their products exclusively and forcing institutions to buy applications that are often more elaborate than required. Given higher education's culture and history of sharing solutions, its institutions stand to benefit greatly from a common framework. Flexibility, stability, and scalability are the ultimate goals of OKI, and its success will be measured in part by the movement of vendors away from "one size fits all" products to commercially viable solutions that meet the diverse needs of higher education. Rather than discouraging or inhibiting commercial development, the ability to plug solutions into any learning system will, it is hoped, encourage the development of commercial solutions that target the specialized needs of different types of institutions and areas of study. Vendors that offer high-quality modules will have access to the entire higher education marketplace, not simply those schools tied to their closed products. Schools and faculty will be able to select from among the best courseware products, assembling their "best of breed."

In other words, most applications need the same sets of functionality, including authentication, authorization, database management, file service, logging, and user messaging. Far more than anyone realized when the Mellon Foundation funded the initiative, the OKI APIs defined in this framework are relevant and important to virtually any commercial or locally developed application that colleges and universities might use.[8] Drawing upon efforts by its partner institutions and others, OKI is defining a "common services" layer that will be a backbone for higher-level service and applications (see Figure 1). Any institutional or commercial application developer will be able to use this set of service definitions and associated APIs. If the world of developers agrees on a common services layer and a set of APIs, anyone will be able to benefit, and the job of developing for various campus infrastructures will be made far easier.

While most of these fundamental API services are not unique to learning management systems, they will nonetheless boost the OKI effort to share learning management applications among institutions and to integrate them seamlessly with existing institutional infrastructures. These basic services will integrate with institutional infrastructures within higher education and permit courseware developers to focus on real pedagogical and administrative issues without having to reinvent underlying functionality or worry repeatedly about the basics, such as how to authenticate a user or how to describe where to store documents and metadata.

The second, "higher" layer of the OKI architecture shown in Figure 1 relates to functionality of particular importance to application developers. Learning applications include student assessment, authoring, collaboration tools, and support for course management. Central administrative systems include student information systems, human resources, and directory management. Academic systems include library information systems and digital repositories of research and education materials.

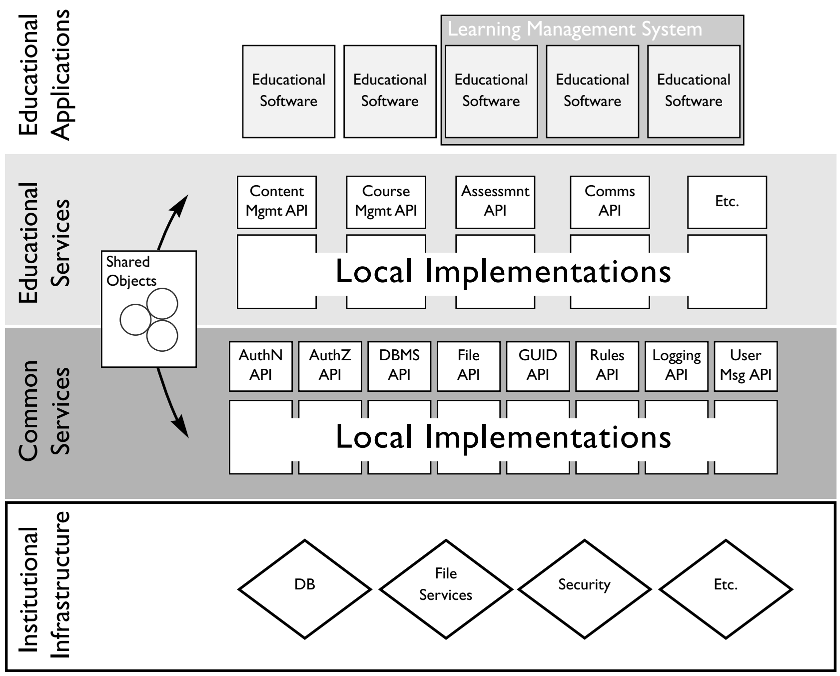

Figure 1. OKI APIs fall in two categories: Common Services and Educational Services. These shared objects form a middleware layer between the physical, institutional infrastructure and each institution's set of educational applications.[9] Copyright © 2002 Eduworks.

Efforts at this level aim to ease administrative tasks—such as course registration, management of class membership, posting of grades, and online examinations—and more advanced services, including publishing, protection of intellectual property rights, control and distribution of course materials, and time-based annotation of streaming video. The efforts also touch on a wide range of user interfaces, from the simple maintenance and delivery of content to "tele-presence," the remote control of lab equipment, and immersive simulation experiments.

In addition, OKI participants will identify, design, and package a set of Web-enabled learning components for use in the widest possible range of educational environments. Working within a standards-based framework will permit colleges and universities, as well as commercial vendors, to construct additional components and modules that can be seamlessly integrated with the set of base components. Examples of such new components include voice recording, instructor voice annotation, multimedia content management, and integration with online data collections.

While the primary product of OKI is the architectural framework specification, MIT and its higher education partners are also building a set of applications that rely upon these interoperable standards as proof of concept of the OKI framework. To demonstrate how the common architecture and common interfaces will permit schools to implement more easily components developed by other organizations, OKI will also make the MIT and Stanford learning management environments (Stellar from MIT and Coursework from Stanford) available as open-source code to anyone interested in running them, including commercial vendors. Other institutions will be able to use these implementations as-is or build upon them. Naturally, the hope and expectation is that any additions will be freely shared with (and thanks to the OKI framework, usable by) other institutions.

MIT will also deliver "OKI in a box," a stand-alone implementation of the OKI APIs on top of a proven infrastructure (in this case, a Linux deployment), all delivered to a campus as a single package. With this product, a campus could begin developing and deploying applications that use the OKI APIs without regard to the complexities of their local infrastructure. Initially, OKI-compliant applications would use the APIs within the "OKI in a box" infrastructure. Once the APIs were implemented to operate on top of the existing campus infrastructure, the applications could be moved without change. (For that matter, they could be moved to any campus running OKI-compatible middleware.)

Given the potential importance of the framework for shared development of all kinds, it is important to create implementations of the APIs for as many existing campus infrastructures as possible. It is also important to have as many application developers and content providers (JSTOR, ArtSTOR, and the like) as possible use them.

The many benefits to OKI's modular approach are interrelated:

- Learning technologies appropriate for a range of teaching and learning requirements will be integrated in a common environment.
- Learning technologies and content can be easily shared among and within institutions.
- Modular construction makes such technology more stable and more reliable and helps assure adaptability to multiple and evolving standards.

- OKI-based systems will scale to the demands of institutions with thousands of faculty and courses, and even multiple campuses and remote access.
- Faculty will gain access to a rich variety of easy-to-use courseware modules developed throughout higher education.
- The amount of time required of faculty to create courseware will be reduced substantially.
- It will be easier to create, customize, modify, and sustain courseware, and thus the cost of development will be lowered and fewer incentives will be required to entice faculty to become involved.
- Systems will cost less to support.
- By making it easier to share courseware, it will also be easier to see whether a faculty member's software/ courseware contributions are actually being used. Colleges and universities would thereby gain at least one measure of the value of such efforts.

Real challenges lie ahead for OKI. The OKI design will need to scale to handle the demands of large institutions with many students, faculty, and nonstandard courses on multiple campuses. We will need to create an architecture robust enough to accommodate diverse environments and future changes in both technology and pedagogy. A permanent consortium of higher education institutions or an independent organization may have to be created to manage and support the evolution of the middleware layer. Perhaps the greatest challenge of all is generating a collective interest throughout higher education to make this happen, and building a developer community to support the framework and create pedagogically interesting applications and updates.

Examples of Applications Using Middleware

The following projects provide examples of educational and administrative applications using middleware: OpenCourseWare (OCW), ArtSTOR, and uPortal.

OpenCourseWare

The Andrew W. Mellon Foundation and The William and Flora Hewlett Foundation are supporting another MIT-led effort, OpenCourseWare. OCW is a large-scale World Wide Web initiative to establish free access to the full breadth and depth of the MIT curriculum via the Internet.

OCW will support noncommercial access to educational materials for all MIT courses (from first-year lectures through graduate seminars) including the syllabi, reading lists, detailed lecture notes, assignments, examinations, problem sets and solutions, and, as appropriate, examples of student work. Students, faculty, and others throughout the world will freely gain access to comprehensive, organized, indexed, educational materials from more than 2,000 MIT courses. OCW will create a consistent path to a full range of high-quality teaching materials in all teaching formats, from traditional lectures and seminars through laboratory-based courses and semester-long projects.

OCW primarily focuses on delivering educational content, although textual content could be delivered without an elaborate courseware platform. Using Stellar (MIT's OKI-compliant learning management system) for content delivery, OCW will be able to incorporate OKI-compliant simulations and other courseware applications such as class discussion lists. The incorporation of OKI-compliant tagging will be important for institutions and individuals who want to make the most of the content shared. Any campus that would like to make its own content available and does so in a way that is compatible with OCW could use any other OKI-compatible learning management system.

The OCW initiative far exceeds the scope of other distance-learning initiatives by offering access to the full MIT curriculum and marshalling the entire campus community behind the effort. Faculty at other institutions of higher education will gain free access to the MIT curriculum. Materials will be posted in ways that will encourage faculty to modify MIT course materials if appropriate attributions are made. University students at other institutions will find materials that complement and supplement their own assignments. High-school students will be able to extend substantially their existing curriculum and take on new challenges. College graduates and professionals who want to maintain existing or explore new interests will gain free access to a web of knowledge and learning. MIT will also make the materials available in formats accessible to users with visual and other disabilities.

OCW will be of special value in developing countries. College instructors there typically have higher teaching loads, more limited experience, and more limited access to key materials. Access to MIT's curriculum has the potential to improve teaching and learning throughout the world. OCW will encourage the translation of its materials into other languages for such noncommercial uses.

MIT has stressed that OCW has the potential to open a dialogue on education by assembling an environment in which university curricula can be studied and compared. MIT expects that OCW will help transform how other institutions of higher education disseminate knowledge and will engage faculty at MIT and elsewhere in providing significant, not-for-profit, educational outreach.

OCW's approach to intellectual property will assure the free distribution of the course materials. Faculty who prepare course materials by themselves will retain copyright and will grant MIT a perpetual, nonexclusive right to use and distribute the materials for noncommercial purposes. If students and staff are involved in the preparation of the materials, MIT will hold the copyright, and faculty will gain perpetual, nonexclusive rights to use the materials. Students will hold copyright over their course work, which will be placed on the site only with their permission.

The project may well inspire other institutions of higher education to follow MIT's lead to modify copyright policies in order to share freely the full range of their own educational materials. Consistent methods for tagging and classifying data should encourage broad participation and enhance the usefulness of multi-university collections.

ArtSTOR

A parallel to JSTOR, ArtSTOR aims to create, maintain, and distribute an electronic library of digital images of works of art, architecture, cultural objects, manuscripts, and related scholarly materials. Working closely with museums, archives, libraries, publishers, scholars, and teachers over many years, the project aims to assemble a series of digital image collections, each with enough depth and breadth to be genuinely useful to scholars and instructors.

ArtSTOR is in the process of creating its image library and has begun user testing with Insight software from Luna Imaging Inc. as one way into the collections. This feature-rich program permits users to search for, display, and annotate images, and to create and save presentations. Insight currently requires users to access its capabilities through a client program downloaded to the user's computer or through a standard Web browser. However, many campuses and museums have already begun using other homegrown or commercially acquired image management systems. They would naturally prefer to access ArtSTOR content the same way they access their local content, using their existing systems. By adding the appropriate OKI APIs (such as authentication, authorization, and digital repository) to the ArtSTOR server, any campus or museum system that implements the same OKI APIs will be able to authenticate its users to the server and access ArtSTOR content.

ArtSTOR might also benefit from the OKI APIs when a feature of an image management system depends on the existence of certain campus infrastructural capabilities. For example, Insight uses the concept of a "group" to permit sharing of presentations, search results, subsets of image collections, and so forth. On a campus, groups may be defined in many ways. They might be created dynamically by the end users themselves through the local file system, be predefined in a course management system to correspond to class enrollments, or reflect an administrative hierarchy. It would not be feasible for Insight to understand the various ways that campuses create, maintain, or access group definitions. Once the applicable APIs are implemented for a campus infrastructure, however, Insight can offer group-dependent features by including the OKI API that contains the "group" abstraction, and then leave it to the middleware to translate to the campus-specific implementations.

uPortal

On the administrative computing side, JA-SIG (Java in Administration Special Interest Group), a consortium of colleges and universities, is developing and distributing an open-source portal. uPortal is essentially a Web application that will provide a single, personalized electronic gateway to all university information and services, including academic content, all administrative applications, and Web-based communications services. All members of the university community can use the portal to perform routine academic and business transactions. uPortal permits users to log on just once to obtain authentication and authorization to all appropriate information resources and applications.

Personalized portals were popularized for the mass market by Web sites such as Excite and Yahoo. However, such sites organize topics in a broad structure that rarely satisfies scholars and specialists. The higher education sector would greatly benefit from the development of tools and services more directly oriented to the scholarly community. More discipline-specific categories and more detailed and personalized structures are needed to assist scholars and students in locating and keeping track of the materials they require.

JA-SIG was formed in 1999 with financial support from Sun Microsystems. uPortal version 1.0 was developed through the collective input from the participating institutions[10] and is based on an open-source model of development that permits the product to meet the needs of a diverse audience at relatively low cost and with a high degree of local customization and control. The project is expected to create a collaborative alternative that will result in free, open-source portal software. As a result, institutions of higher education and scholars will be able to consolidate and customize online information in ways that are fully appropriate locally without having to rely on closed, proprietary solutions.

Such development efforts are beyond the means of most colleges and universities. Funding from The Andrew W. Mellon Foundation will allow completion of the development of version 2.0 of uPortal, packaging of the software for wider distribution throughout higher education, and establishment of a long-term portal support program. This collaborative effort will leverage the skills of software developers at many institutions of higher education and provide an effective solution to common needs. The project will tap the expertise of the 30 member institutions of JA-SIG. Finally, the development of the new, object-oriented, uPortal technology will promote collaboration among institutions and encourage the development of a common infrastructure while permitting institutions and individual scholars to customize their approaches and share new techniques and software enhancements.

A new "Portlet" standard to facilitate the creation of specialized modules that will plug into uPortal or any other compatible portal system is also emerging. Portlets will interface with other applications through a single Web interface that will be centrally administered yet customized and personalized to meet individuals' personal and business requirements.

The uPortal and Portlet efforts underscore the need for a comprehensive OKI-like approach to administrative software. Higher education needs to develop a model for sharing administrative applications that will decrease the cost of ownership, continue to offer enhanced functionality to the community, work with the range of campus infrastructures, and keep current with new and emerging technologies.

Conclusion

The Greek mathematician Archimedes claimed that with a sufficiently large lever, he could move the earth. Unlike other sectors—such as airlines or banking—where competing institutions need to protect their individual competitive advantages, higher education's institutions have and use a lever in their willingness to share solutions to common problems. By sharing objects and modules and by working within collaborative environments, higher education can lower the threshold of effort and therefore the costs to develop and maintain instructional and administrative technologies.

We are off to a good start. A groundswell of enthusiasm followed MIT's announcement of OKI and OCW. One of the main commercial course management system vendors, Blackboard, announced in early 2002 that they would incorporate OKI APIs into a future release.[11] The University of Michigan has adopted the OKI framework for its new learning management system, CHEF (CompreHensive collaborativE Framework), and other campuses are moving in the same direction.

Our campus infrastructures will never resemble one another precisely. Through the use of middleware, however, institutions will be able to assemble and run a common architecture and interfaces that will substantially decrease the cost of application development and ownership, offer high functionality and a greater range of applications to the whole higher education community, and permit institutions to remain current with new and emerging technologies—all within a sustainable and affordable long-term support and maintenance model. To lock in our new era, we need only a large enough lever to move a critical mass of developers and the decision makers within higher education to collaborate and cooperate in building a common architectural foundation.

NOTES

1. Natalie Angier, "Why We're So Nice: We're Wired to Cooperate," *New York Times,* July 23, 2002, p. B-1.

2. Tim Berners-Lee, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor* (New York: HarperCollins, 1999), p. 28.

3. Electronic Frontier Foundation, *EFF Whitepaper: Unintended Consequences—Three Years under the DMCA*, http://www.eff.org/IP/DMCA/20020503_dmca_consequences.html.

4. Kerberos is a commonly used network authentication protocol developed at MIT.

5. A third important middleware effort, the Internet2 Middleware Initiative, is described comprehensively at http://middleware.internet2.edu/. The effort, which is using the National Science Foundation Middleware Initiative as a testbed, is moving toward the deployment of core middleware services at Internet2 universities.

6. The institutions collaborating with MIT are Stanford University, Dartmouth College, North Carolina State University, University of Michigan, University of Pennsylvania, University of Wisconsin–Madison, University of Washington, Indiana University, and the University of Cambridge.

7. Learning management systems have become a core component of most campus's information technology infrastructures. The 2001 Campus Computing Survey reports that nearly 75 percent of all participating institutions had a standard campus course management system and that nearly 20 percent of all courses were supported to some extent. See the 2001 Campus Computing Survey on the Web at http://www.campuscomputing.net/pdf/2001-CCP.pdf.

8. The original Andrew W. Mellon Foundation grant meant to solve the problem of defining one particular application: learning management systems that met higher education's functional needs and would integrate easily with most campus infrastructures. Blackboard, WebCT, and other commercial products simply do not satisfy these criteria. Some institutions with extensive resources had built their own systems, honed to work well on their campus infrastructures. MIT (and Stanford) proposed developing an architectural framework to develop learning management systems that could be shared in part or whole.

9. Eduworks Corporation, "What is the Open Knowledge Initiative? A White Paper," July 24, 2002. Available on the Web at http://web.mit.edu/oki/learn/whtpapers/OKI_white_paper_120902.pdf.

10. Institutional participants include the University of Delaware, Boston College, Stanford University, Dartmouth College, University of Michigan, University of Virginia, Yale University, Cornell University, University of Washington, University of Minnesota, Brown University, Princeton University, Georgetown University, the University of British Columbia, and more than 15 other institutions.

11. The May 20, 2002, press release stated, "Blackboard Inc. announced today a broad strategy to adopt industry standard APIs (Application Program Interfaces) from OKI within the Blackboard e-

Education Suite. Specifically, Blackboard's Building Blocks open architecture will base future releases on key OKI specifications, enabling a broader variety of 3rd party applications to work seamlessly with Blackboard. This announcement represents an important endorsement of the emerging OKI specifications and will help accelerate OKI's status as an industry standard in the higher education market. Through their relationship as common members of the IMS Global Learning Consortium, Blackboard and OKI institutional partners are working together with other IMS members to help define the next generation of interoperability standards for educational technology."

Ira H. Fuchs is Vice President for Research in Information Technology at The Andrew W. Mellon Foundation.