The benefits to colleges and universities of collaborating on information technology are enormous and far exceed the drawbacks and compromises institutions may need to make to work together. 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 cost-effective use of information technology. Ira Fuchs, vice president for research in information technology at The Andrew W. Mellon Foundation, discusses the impediments to collaboration as well as remarkable current efforts that may point the way to the future of information technology in higher education.

The Importance of Collaboration

The use of technology has spawned invigorating and enriching transformations in teaching, learning, and research over the last 20 years or so. However, these efforts, which are being duplicated throughout institutions of higher education, often involve a dramatic increase in workload for faculty and technical support staffs. Collaboration can significantly reduce the institutional workload required to create and maintain educational software. On the administrative side, most institutions have struggled with implementation of new applications, which have substantially increased the cost of system ownership. The aggregate cost of the academic and administrative changes across all of higher education is enormous: Universities such as Stanford, MIT, and Princeton have already spent upwards of $100 million each to replace a mere subset of their administrative systems.

Hence, the most cogent reason for collaboration is cost and time savings, particularly in administrative applications where a generic approach may satisfy most institutions. Collaboration on courseware and instructional applications offers the additional benefit of tapping the creativity and ingenuity of faculty and staff at many institutions. A generic approach to courseware may simultaneously lower the
cumulative costs of development and result in a far richer variety of well-developed materials from the higher education community.

Impediments to Collaboration
There are many obstacles to improving information technology collaboration among colleges and universities, yet all can be overcome by a strong commitment to do so on the part of a critical mass of institutions. Those obstacles include the following:

- **Heterogeneous technology infrastructures.** Applications written for one campus are unlikely to run at another campus without first undergoing significant and costly modifications.

- **Application development is local and expensive.** 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.

- **Intellectual property rights.** Today’s copyright laws are such that intellectual property issues have become major obstacles to the creation and sharing of instructional materials.

- **Inadequate faculty incentives.** Decisions about advancement at most institutions continue to depend overwhelmingly on research output. Few ways have been identified to reward faculty who invest time on courseware development.

- **Teaching is a customized, personal endeavor.** Using courseware developed by another faculty member is a dilemma due to the historical “personal” nature of teaching. Further, until recently, faculty haven’t had the technical tools to integrate materials developed by other faculty into their own courses.

Middleware
Notwithstanding all the barriers, there is hope. We can substantially lower the cumulative threshold of effort and therefore the costs required to develop and maintain instructional and administrative technologies by sharing software and modules and by working through collaborative development environments. The key to success—the basis of our hope—is middleware.

There are myriad definitions of middleware. Most agree, however, that middleware is the layer of software that stands between the campus infrastructure (campus 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 problematic, direct communication between the applications and the infrastructure and to provide safe translations between the layers. Appropriately maintained, the middleware layer will reliably translate requests and pass along information and instructions that the physical infrastructure and individual applications will understand. As a result, 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), database access, directory services, content storage and retrieval, file sharing, and access to campus external network connections.

The primary feature of middleware is its 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 work with both our varied campus infrastructures and the applications that run on those infrastructures. An API for file sharing, for ex-
ample, would need to work with the different file sharing infrastructures that exist on our campuses.

Middleware must evolve constantly to accommodate developers and advances and modifications in local infrastructures. 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, higher education will need a critical mass of both 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 more scalable and will provide consistently high performance. Support services and maintenance will be simplified, and faculty will be much more likely to produce courseware if the tools at their disposal vastly reduce the effort required.

The Open Knowledge Initiative
The Open Knowledge Initiative (OKI), funded for its first two years by The Andrew W. Mellon Foundation, is an ongoing effort on the part of MIT and a group of collaborating, partner institutions to design and develop an architecture for course management systems that is open-source, flexible, scalable, and sustainable. The project will facilitate the development and delivery of learning management systems and educational software applications. Today, learning management systems have become a core component of most campuses’ information technology infrastructures.

By clearly defining points of interoperability, the architecture will allow the components of complex learning environments to be developed and updated independently of each other. New components could be plugged into the learning infrastructure using OKI’s standardized APIs. Any institutional or commercial application developer will be able to use the OKI-defined set of “common services” and their associated APIs. 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 basic underlying functionality or worry constantly about basics such as how to authenticate a user or describe where to store documents or metadata.

Given the potential importance of the shared framework for development of all kinds, it is crucial that applications of the middleware’s APIs be created for as many existing campus infrastructures as possible, and that as many developers and content providers as possible use the APIs so that a critical mass for collaboration can form.

Examples of Applications Using Middleware
The OpenCourseWare (OCW) initiative, supported by The Andrew W. Mellon Foundation and The William and Flora Hewlett Foundation and led by MIT, is a large-scale effort to establish free access to the full breadth and depth of the MIT curriculum via the Internet. 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 formats, from traditional lectures and seminars to laboratory-based courses and semester-long projects.

MIT’s learning management system, Stellar, which will be used for content delivery, is OKI-compliant. Any campus that wants to make the most of the OCW content or that wishes to make its own content available and does so in a way that is compatible with OCW will also be able to use any other OKI-compatible learning management system. The potential of such broad accessibility to increase access to learning materials is enormous. OCW
will be of special value in developing countries in particular; access to MIT’s curriculum has the potential to improve teaching and learning throughout the world. MIT expects that OCW will help to transform how other colleges and universities disseminate knowledge and that it will engage faculty at MIT and elsewhere in providing significant, not-for-profit, educational outreach.

JSTOR, the digital archive for back issues of journals, and ArtSTOR, the developing digital archive of images of works of art, architecture, cultural objects, manuscripts, and related scholarly material, also greatly enhance their accessibility by using OKI’s APIs. By adding the appropriate OKI APIs (e.g., authentication, authorization, and digital repository) to the ArtSTOR server, any campus or museum system that implements the same OKI APIs will be able to, for example, authenticate its users to the server and access ArtSTOR content.

Perhaps the greatest challenge we face is to generate a critical mass of developers and decision makers within higher education to collaborate and cooperate in building a common technological foundation...

Conclusion
The separation of educational software applications from supporting institutional infrastructures enabled by middleware creates a modular approach with many interrelated benefits. Although campus infrastructures will never resemble one another precisely, middleware enables institutions to assemble and run a common architecture that lowers the threshold of effort and therefore substantially decreases the cost of development and ownership. Middleware offers high functionality and a greater range of applications to the entire higher education community and permits institutions to stay current with new and emerging technologies—all within a sustainable and affordable long-term support and maintenance model.

Perhaps the greatest challenge we face is to generate a critical mass of developers and decision makers within higher education to collaborate and cooperate in building a common technological foundation that will be more stable and less costly, and that ultimately will enrich teaching and learning around the world.

Ira Fuchs is vice president for research in Information technology at The Andrew W. Mellon Foundation. Fuchs is also senior technology advisor to the president of Princeton University, and chief scientist of JSTOR. He was vice president for computing and information technology at Princeton University from 1985 to 2000.