



# Having Your

By Bill Olivier

# CAKE

## and Eating It

### The e-Framework's Service-Oriented Approach to IT in Higher Education

**O**ver the last ten years, the use of the Internet has moved from a small number of specialists to the population at large, including all ages and social groups. More recently, for those with access to computers and broadband, the Internet has become almost ubiquitous, with people using it at work, on the move, and for all sorts of leisure and personal activities.

*Bill Olivier is Development Director for Systems and Technology for the U.K. Joint Information Systems Committee (JISC).*





## The new software-design paradigm focuses on the use of small components that can easily exchange data with each other and that can be mixed and matched.

Much of this use of the Internet has been made possible by new approaches to systems design. These approaches involve the development of Web services that can be combined in a wide variety of ways, in a mix-and-match approach. In contrast to the model in which IT companies build large, monolithic systems, developers are increasingly creating small, clever tools that can be shared and joined in ways that produce new, flexible applications.

The research and education world is currently experiencing the impact of many new drivers of change, partly enabled and stimulated by the possibilities offered by the Internet. The researcher or learner who moves between various groups and institutions and who works in a number of different countries is now commonplace. Yet these possibilities are often framed in a context in which the technical systems within one institution cannot easily “talk to each other” and share information, let alone communicate across institutions, jurisdictions, or national boundaries.

A new context is needed for higher education: a world in which technology plays a role in supporting people in whatever they want to do; a world in which technology is flexible, responsive, and easily adaptable to needs; a world in which technology no longer determines, or constrains, what people can do. The new software-design paradigm focuses on the use of small components that can easily exchange data with each other and that can be mixed and matched. This is the essential characteristic of a *service-oriented approach*.

To apply this approach to the higher education context, we need to look more closely at the issues. Last year, Ithaka published a report on a widely researched study into the need for, and potential nature of, an “Organization for Open Source Software” (OOSS). The authors began by

examining academic leaders’ concerns with current software provision to higher education:

We found a considerable amount of evidence attesting that many college and university leaders are dissatisfied with the cost and performance of software, and that this is a matter of significant concern to them. The areas of dissatisfaction can be grouped under three headings: (1) Cost. Many institutions have spent millions or even hundreds of millions of dollars implementing and customizing administrative systems, with significant costs incurred each time the vendor phases out old versions of the software. (2) Performance. Many commercial products are not well tailored to the needs of higher education, and because they are proprietary it is difficult and expensive to make the desired modifications. (3) Control. College and university leaders are concerned that consolidation in the sector may result in commercial software vendors having unfair pricing leverage in their negotiations with the higher education (HE) community. The areas of greatest dissatisfaction and concern to senior leadership were in the categories of administrative software, both those that are not specific to HE (financials / purchasing / physical assets / space management), and those that are specific to HE (student administration / financial aid / admissions / registrar / grants management). They are also concerned about course management systems, which are seen as core to the academic mission.<sup>1</sup>

The *service-oriented approach* (soa), based on open service interfaces, can also be part of a solution to these issues of cost, performance, and control. The proposed Quali Student Services System project, for

example, combines a community-source approach to organization and licensing with a service-oriented approach to architecture: the two approaches are complementary, rather than contradictory.

### The Vision

To understand how the soa can be part of a solution, one needs to have some idea of the vision that the service-oriented approach is intended to realize. In essence, the soa seeks to separate out the core functionalities needed by a number of different applications into a series of modular components, provided as loosely coupled services. Authentication and authorization are examples that are almost ubiquitous and that are already beginning to be provided as separate services. Additional candidate services are people management, resource and content management, and workflow management—common functions found in many different applications. Much of the code in current software applications is redundant, performing the same basic tasks (e.g., file/data management, security, business logic, user interface) that are performed by equivalent code in other applications. Even code that is specific to a vertical domain may be shared by other applications in the same domain. Thus, only a relatively small part of the code may be unique to an application. This redundant code costs money to create, and it costs even more money to maintain. By separating out these functions into reusable services, the soa can drastically reduce the total amount of code that an enterprise running hundreds of such applications needs to support. Thus the claim is: less code equals lower purchase costs, and fewer support staff equals lower running costs.

Once separated out into services, the functions or tasks need to be provided with an agreed-upon open interface, and the wider the agreement is on these tasks, the greater the market for such services and the lower the price. (Considerable time and effort of experienced and technically skilled people is needed to establish such agreements, but several e-Framework partners, with both long-standing and long-term commitment, significantly fund activities that



## The creation of an adaptive and flexible IT infrastructure that supports the evolving organizational goals is another key element of the soa vision.

contribute to the development and adoption of open specifications and standards.) Such services can be implemented once and then accessed by different applications as and when needed. In contrast to traditional approaches—where, in response to the diverse needs of higher education, each vendor integrates ever-increasing amounts of enterprise functionality into one large monolithic application—the soa features a multiplicity of services that can be called on and integrated by smaller, lightweight applications, which can be developed more quickly in response to evolving needs and priorities. The services supplied behind the open interfaces can differ in terms of scalability, performance, and other characteristics, enabling an institution to balance the infrastructure according to its particular needs, rather than being forced to accept whatever generic compromise was built into the application. This in turn enables the development of an ecology of smaller, specialized suppliers.

The lightweight “composite” applications, which sit above and call on these services, are much easier and quicker to develop—and therefore also to adapt or change. Critically, they also allow greater diversity in the software that is provided to support “the way things are done” while building on standardized and commoditized services. Although the underlying, common services may be centralized, a properly designed service-based approach gives the schools and even departments *greater* control and autonomy over the applications they need and at the same time permits institutions to retain central control over those aspects of the system that need it, such as system stability and policy compliance, as well as over core processing and information services. As with balancing needs and budget, the soa permits institutions to negotiate their own, distinctive balances of autonomy and centrality. Over time, this local control

of domain-specific functionality, coupled with the greater ease of developing and changing software at this level, should make it easier and cheaper to support the steady improvement of working practices and processes, in addition to being able to integrate new services and applications. The creation of an adaptive and flexible IT infrastructure that supports the evolving organizational goals is another key element of the soa vision.

When applied in any given organization, the service-oriented approach (soa) will result in a service-oriented architecture (SOA) or, as some now call it, an enterprise service architecture (ESA), tailored to support the unique strategy of the organization in its unique context. Christopher J. Mackie, of the Andrew W. Mellon Foundation, has noted: “The sheer number of applications that a higher education enterprise requires is burdensome already, and with new and expensive-to-sustain ‘cyberinfrastructure’ arriving, there are serious IT budgetary/capacity challenges upcoming for all but the wealthiest institutions. Enterprise SOA offers to cut that burden considerably, by refactoring redundancy out of the institutional code-base and thereby reducing the total amount of code that an institution must maintain. In conjunction with community-led development processes, it promises a better, longer-term strategy for coping with the looming higher education software-maintenance challenges.”<sup>22</sup>

### Applying the Service-Oriented Approach to the Issues of Cost, Performance, and Control

Returning to the issues raised in the Ithaka report and applying the soa suggests the following:

1. *Cost* can be addressed initially by identifying common functionality and making this available from existing

systems by providing them with Web service interfaces. This acts to increase their value by making the information that is “locked up” in these systems more widely and easily available. When an old system needs to be replaced, as long as the replacement system and services can be provided with the same open-service interfaces, then applications that worked with the old systems will work with the new system. If new functionalities are required, the option exists of providing these as separate common services, thus paying for them only once. In either case, system integration and switching costs are reduced through open standards.

2. *Performance*, in the above-quoted sense of systems needing to be better tailored to the requirements of higher education and easier to modify, can similarly be addressed. Once core information and functionality is made available through common services, applications in the lightweight service-integration layer can be easily developed and subsequently changed, thus supporting unique needs and ways of working. This task can be facilitated by using the growing number of graphical tools for defining, and then generating, the processes needed to coordinate the underlying services.
3. *Control*, in the sense of concern over the consolidation of suppliers into monopolies or oligopolies, would be addressed, eventually, by having a much more modular infrastructure, with no single supplier of large monolithic systems. Further, if common needs, addressed by common services, can be established internationally, a significantly larger market can develop, supporting a larger number of software suppliers.

However, the Ithaka study also noted that academic leaders expressed a number of concerns about depending on open-source software. A further advantage of the service-oriented approach is that, providing open standards are adopted, commercial software and open-source software can work together and can be mixed and matched according to needs, budgets, and the performance of



## There is a need for a map of what has been developed and for the underpinning standards and specifications.

the software. Open-source software that is no longer supported can be replaced by a commercial product, just as the product of a commercial company that no longer serves the sector or that ceases operation can be replaced by open-source software.

Granted, the approach of putting service interfaces over existing monolithic systems has not always been smooth. Services surfaced in this way can be difficult to provide reliably; authority/responsibility for them can be too diffused (assuming the underlying applications are in departmental silos); and critical dependencies can be easy to introduce and difficult to manage. What if the most-desired service sits on a departmental server that is ancient and underpowered? What if someone decides to shut down the server—how would that person discover every dependency the server anchors?

A danger can arise here: a technical remedy might be applied with too little attention to the organizational, social, and political dimensions of the business process model. This in turn suggests that to fully realize the benefits of the service-oriented approach, institutions must take the wider context into full account. Who will be using the available services? And what will their demand level be? In other words, it is necessary to also start developing an enterprise-wide approach to the development of a common service-based middleware architecture, on which the lightweight, agile, “composite” applications can be built.

One of the underlying issues for both commercial and open-source software providers is scale: both types of providers depend on defraying costs across a large (enough) user base, whether at the sales end or at the developer-community end. For either to flourish in a service-based world, a key task is identifying common functions and information requirements that form the basis for services. As suggested above, this is the biggest task,

and the biggest risk, facing the service-oriented approach to providing higher education with systems tailored to its needs. The task is large because it involves identification, comparison, and distillation of common requirements across the sector and, to truly reduce costs, across the sector globally and across other sectors. The risk is that the more widely the net is spread, the less commonality there is to be found. This has not been a problem in other areas, but higher education may be different—and there may still be the need to compromise a little and converge.

The task is not just suited to but positively demands collaboration. This is where the international e-Framework for Education and Research comes in.

### The e-Framework

The e-Framework for Education and Research (<http://www.e-framework.org/>) has been established to help the research and education world take advantage of the opportunities offered by the service-oriented approach. An initiative of the U.K. Joint Information Systems Committee (JISC) and the Australian Department of Education, Science, and Training (DEST), the e-Framework was undertaken as part of their Cooperation Framework Agreement. The e-Framework partners are JISC, DEST, the New Zealand Ministry of Education, and SURF (the Netherlands). The e-Framework partners, as national bodies responsible for funding IT development, are able to influence the policy and development of educational and research infrastructure at a national level. The primary goal of the e-Framework is to facilitate technical interoperability within and across education and research through improved strategic planning and implementation processes.

Much valuable experience has been gained by all those involved, and a great deal of technical development has taken place over the last few years. Now that this approach is becoming more mature,

both in this context and also in the wider software and business world, a number of issues have begun to emerge. There is a need for greater coherence in development—for a map of what has been developed and for the underpinning standards and specifications. Such information will enable a strategic approach to planning development programs and will provide institutions with information on what is under development and what is available and ready for adoption and mainstream use.

As noted, the e-Framework draws on two key technology developments: the service-oriented approach and Web services. Its Web site or knowledge base currently comprises technical information on open services and service compositions, or reusable “service usage models,” designed to provide a given technical function. The plan is to extend this with a higher-level knowledge base with sets of domain maps, scenarios, and practice and process models. Both the technical and the higher-level knowledge bases will be supported by materials in the form of guides, methodologies, and analysis. The e-Framework is guided by several underpinning principles:

- A service-oriented approach to system and process integration
- A commitment to open standards
- A recognition of the central importance of community involvement
- The need for open and collaborative development activities
- The deployment of these approaches in a flexible and incremental way

The benefits of the e-Framework can be seen from several perspectives. For institutions, the e-Framework will enable

- the alignment of strategies and infrastructure development to support the areas of education and research;
- more choice of systems and suppliers;
- more freedom in terms of buy/build/borrow/outsourcing decisions;
- an improved return on investment in existing systems;
- more effective communications between communities through shared understanding; and



## The e-Framework will provide a map of the territory, but it is unlikely that any institution will deploy the whole of the framework.

- interoperability within and across institutions and national boundaries.

For developers, the e-Framework will support

- a better understanding and dialogue between suppliers and customers;
- more rapid development cycles through reusable components and hence a faster response to customer requirements;
- greater opportunity for the entry of small innovative players into the market;
- better communication and collaboration among developers themselves; and
- more flexible business models for software development.

For partners, the e-Framework will provide

- a map of a complex environment;
- a strategic planning tool for prioritized investment in standards development and prioritized investment in interoperability technologies; and
- an improved return on investment through coordination and collaboration among partners.

More generally, the e-Framework will provide information about the following:

- Services, the available open specifications, and standards
- “Service Usage Models” (SUMs): compositions of services that perform a specific function
- Links to background information on use, good practices, embedding process models, design, context, and available implementations
- Guidance on many aspects of identifying common services, from high-level common inter- and intra-domain

needs to low-level common application functionality

The e-Framework will provide a map of the territory, but it is unlikely that any institution will deploy the whole of the framework. The more likely scenario is that an institution will incrementally and agilely use those elements in the e-Framework knowledge base that address its needs at a particular time. Rather than replacing existing or legacy systems, it is intended to provide guidance on how these systems could be integrated by adding Web service interfaces to specific parts of their functionality, thus deriving greater value from them. However, this has to be done with an appropriate balance of centralized common services and decentralized application development. In turn, good practice recommends that the incremental approach of adding service interfaces to existing systems be undertaken in the context of the development of an institution-wide service architecture. This architecture can also be developed using an agile approach, first establishing a broad overall architecture, but filling in the detail only as demanded by the priorities of implementation projects. This is necessary to ensure that a purely incremental approach still realizes the benefits of common services, rather than resulting in an incoherent set of similar and overlapping services, each with different interfaces.

There is a great deal more to be said, and discovered, about successfully implementing a service-based architecture. Currently, the e-Framework Web site offers a sketch of what is required. It will be open to contributions and freely available for use. Its knowledge base is intended to provide a tool for thinking, planning, and coordination—a strategic tool that can support the international education and research communities in their exploitation of the next generation of technology

development. However, an enormous amount of work remains to fill out the details. This is why developing collaboration and international partnerships is crucial to the progress of the initiative.

A participant at a recent e-Framework workshop stated: “Every dean runs things their own way and will not accept any software solution that forces them to run their schools in a uniform way. Flexibility is essential.” What is true within an institution is equally true *across* colleges and universities. A key issue, then, for those who are developing software for higher education—whether in-house, commercial, or open source—is how to address these diverse needs and approaches without incurring the huge costs involved in producing a unique system for each institution (let alone for each school within an institution) using traditional approaches. In their funded projects, e-Framework partners are exploring new ways of enabling stakeholders and developers to work together in adapting and implementing the service-oriented approach. Our hope is that the e-Framework will help provide coherence at the technical level, as well as some guidance and further references on how to go about developing institution-wide service-based architectures.

What all institutions would like is innovative software tailored to their unique needs (“the cake”) at a commodity-level price (“and eating it”). The extent to which the service-based approach can provide this is unclear at present. The goal of the e-Framework partnership is to determine how much common functionality can be agreed upon while still supporting the requisite variety needed for institutions and their schools and departments to find the best ways of meeting their goals in a changing environment. *e*

### Notes

Thanks to Christopher J. Mackie, of the Andrew W. Mellon Foundation, and Ian Dolphin, of the University of Hull, who constructively reviewed drafts of this article.

1. Paul N. Courant and Rebecca J. Griffiths, “Software and Collaboration in Higher Education: A Study of Open Source Software,” July 26, 2006, <<http://www.ithaka.org/about-ithaka/announcements/ooss-study-final-report>>, pp. 3–4.
2. Christopher J. Mackie, private communication to the author.