OKI Interoperability

May 4, 2007

Authors:
Ed Walker, Jeff Kahn,
Keith Kiser - Principal Investigator[1]

Sponsors:
The Andrew W. Mellon Foundation,
Ithaka, and the Massachusetts Institute of Technology

[1] We are grateful to Chris Mackie, Meredith Quinn, and Wilson D'Souza for their time, expertise and valuable commentary on earlier drafts. Barnaby Gibson was a helpful guide to us in legal matters.

This page intentionally left blank.

Table of Contents

Executive Summary

The limited interoperability software and content for higher education is a matter of serious, ongoing concern. Many valuable educational resources remain stove-piped to the context in which they were created. Collectively, the resources are a box of parts, rather than an integrated ecosystem supporting learning, teaching and research.

Various approaches to achieving greater interoperability and re-usability are known and available, but no approach has proved to be a magic bullet. Among the available approaches are the Open Service Interface Definitions (OSIDs), developed by the MIT-led Open Knowledge Initiative (OKI). Unlike many interoperability standards and tools, the OSIDs are intended specifically to address interoperability of educational resources; however, their adoption has not been as rapid or as universal as their original sponsors hoped it would be.

In January 2007, Ithaka and MIT worked on an interoperability study, funded by Ithaka, at the request of the Mellon Foundation, to determine the viability of using the OKI OSIDs as an approach to achieve greater interoperability within the Mellon Foundation supported projects. This report is the result of interviewing the people directly involved with those projects to obtain their views on achieving interoperability, asking them whether they believe the OKI OSIDs can be utilized to achieve it, and if so, asking them how best to accomplish that objective.

Our respondents pointed out that not all projects require external interoperability. They acknowledged the general possibility that the OSIDs might be useful for accomplishing interoperability in at least some cases, but argued that questions of whether to use the OSIDs or any other approach to accomplish interoperability made no sense in the abstract, because such questions could only be answered effectively in the context of specific projects with specific deliverables. They further argued that efforts to increase interoperability should only be linked to projects for which interoperability was a key stakeholder requirement, and in particular that the use of the OSIDs to achieve interoperability was likely to work best for projects in which the achievement of cross-functional, cross-institutional or cross-vendor interoperability would require integration of multiple component services in multiple enterprises in a many-to-many relationship. Within those parameters, respondents were also able to agree on reasonable next steps to investigate the development of broader, easier interoperability among current and future Mellon Foundation supported projects. They distinguished two projects, The California State University Digital Marketplace (DM) initiative, and Kuali Student, as particularly suitable platforms for such investigations. Within those projects, they further identified four promising focal areas for interoperability development: Identity Management, Repository Submission and Federated Search, Workflow, and Interoperable Gradebook Reporting.

This page intentionally left blank

1. Introduction and Framing

1.1. Introduction

The degree to which interoperability has been achieved between educational software, academic content and underlying infrastructure is a matter of ongoing concern. The innovative functionality and potential impact of these resources are impressive, but many of them remain stove-piped in the context in which they were created, instead of being adopted across a variety of enterprise environments. Collectively, the resources are a box of parts, rather than an integrated capability for learning, teaching and research.

Educational applications like Sakai, Kuali Financials, DSpace, Fedora, and others were developed with the assumption they would be integrated into various enterprise service environments that include other legacy or new services. This integration phase is beginning now. Because the applications have already been developed, discussions about whether they could have been built to be more interoperable will not be productive; on the other hand, discussions about how to make them more interoperable going forward are imperative, because no enterprise system is an island. The ease of integrating service components or system designs in a new environment is an appropriate measure of their utility and quality, and not providing for re-use by others is inconsistent with an open source, collaborative model of enterprise software development. Pragmatically, too, reducing the delay and cost of integration and interoperability facilitates the adoption and evolution of educational resources.

Various approaches to achieving greater interoperability and re-usability are known and available. No approach is a magic bullet, and the maturity and applicability of approaches varies. Among the available approaches are the Open Service Interface Definitions (OSIDs) developed by the MIT-led Open Knowledge Initiative (OKI). OSIDs attempt to address interoperability as it applies to end-users and developers of educational resources. The underlying basis of the OSIDs is a services-oriented architecture (SOA): they are intended to interconnect various educational and educational-support services into a coherent ecosystem supporting teaching and learning. However, the utility of the OSIDs and other approaches has not yet been adequately assessed, in large part because such utility can only be measured in broadly interoperable production environments—arenas that have not yet materialized in higher education despite the availability of the OSIDs as well as alternative interoperability approaches.

In January 2007, Ithaka and MIT began a study, at the request of the Mellon Foundation, to increase interoperability in the education and research environment. This study was undertaken out of the belief that the higher education community could benefit from the utility of using the OSIDs to deliver interoperability across the enterprise. The study was launched with the following objectives:

- Collect criteria for evaluating the long and short-term benefits, costs, and risks of incorporating interoperability into the design and development of academic technology infrastructure
- Collect specific data on the usefulness of the OSIDs, or lack thereof, in implementing interoperability

- Identify specific opportunities for further work to advance the level of interoperability among components of a SOA in higher education.

More than 30 people from 10 organizations were interviewed for about 60 minutes each, either in small groups or individually. The respondents were asked which criteria they use to select an approach to integrating system components and how they make trade-offs between interoperability features and budget, time and personnel constraints. They were also asked whether they had used the OKI OSIDs, and asked to assess the value of the OSIDs as they had experienced them. Their responses were frank and thoughtful, as well as remarkably consistent.

The respondents in the study were in agreement that the education and research environment requires a high level of interoperability among software applications, sources of academic content, and the networked computing infrastructure of educational enterprises. During the past few years, they acknowledged, SOA has become the favored approach to enterprise system design and development. They further agreed that the appropriate measure for any approach to designing or implementing SOA is utility-in-practice; that is, measuring features in terms of whether they aid in designing, developing, deploying, and maintaining production services in enterprise scale environments. Two key objectives of SOA are to improve the maintainability of the enterprise software environment and to enhance flexibility and adaptability. The OSID interface definitions attempt to improve maintainability and enhance flexibility by facilitating the independent integration, maintenance or replacement of the individual services that make up an implemented SOA. Thus, a key evaluative question for almost all of our respondents was whether OSIDs deliver this utility in practice, as well as in theory. By far the majority of respondents think that the question is still open.

At the same time, despite the consistency of many of their opinions, when our respondents used terms like "interoperability," "performance," "production level," "OKI," "Sakai," "Web Services," "developer," "architect," "standard," and so forth, it quickly became clear that they did so with very different unspoken assumptions about contexts and connotations. The Sakai project, for example, was seen by some as an enterprise application—a learning management system—and by others as a framework for building such applications. Definitions of "production-level" varied widely in scale according to the respondent's institutional frame of reference. Many respondents used "OKI" and "OSID" as interchangeable terms; few clearly understood the distinction between OSID definitions and OSID implementations. Some respondents thought of standards as practices-in-use; others saw them as abstract entities. It was apparent to the interviewers that the lack of shared understandings of these terms has obscured important details and led to communication problems and mistaken expectations, and that it continues to impede collaboration even now. More explicit differentiation between projects and people, as well as among definitions, architectures, implementations, models, standards, etc., can only make collaboration easier. Some of the responsibility for that clarification must fall on the respondents themselves, but it can and should be aided by clear, adequate, and consistent communication by the projects involved.

1.2. Purpose and Scope

The purpose of this report is to provide a consolidated summary of what was discovered from the interviews, as background and preparation for participants in a working meeting on 28 March

2007. The participants in the meeting discussed this report and its findings. Please see Appendix D for meeting notes.

1.3. Organization of the Report

This report presents the conclusions drawn by the investigators from the respondents' answers to our questions on interoperability and the use of OSIDs. Even though the report discusses interoperability in the context of OKI OSIDs, it is well known that OSIDs are not the only ways in which interoperability can be achieved. The goal of this report is not to compare OSIDs directly with alternative approaches, but rather, to assess the effectiveness of the OSIDs from the respondents' point of view. The report allows readers to make their own comparisons with other approaches, as needed.

The body of the report consolidates the interview responses into guidelines for choosing projects that are likely to improve interoperability and criteria for selecting approaches within those projects that will maximize delivered interoperability. Summaries of these guidelines are provided in Appendix B: Choosing a Project and Appendix C: Choosing an Approach.

Section 2 discusses the objectives of the study, who was interviewed, and the questions asked of them.

Section 3 contains a detailed summary of the respondents' answers.

Section 4 discusses areas that projects should examine to improve interoperability, focusing on two projects in particular, Kuali Student and the Digital Marketplace project.

Section 5 summarizes lessons and insights learned from the respondents.

Appendix A includes the complete list of respondents.

Appendix B contains a decision matrix on how to choose a project as an interoperability candidate.

Appendix C contains a decision matrix for evaluating approaches to achieve interoperability in order to maximize Return on Investment (ROI).

Appendix D contains a list of attendees at the March 28 workshop and meeting notes.

2. Methodology

2.1. Study Objectives

The study had the following four objectives:

1. Interview the stakeholders identified by Mellon, Ithaka and MIT in-depth, regarding their individual questions, criteria and constraints when addressing interoperability issues.

2. From the interviews, develop a common set of decision criteria and measures of utility for choosing projects in which to apply a particular interoperability approach.

3. Gather enough information to facilitate discussion during the March workshop on the parameters and conduct of appropriate interoperability assessment experiments.

4. Develop summary talking points and a starting-point experimental framework – hypotheses, tests, and metrics – for discussion and decision-making.

To accomplish the study objectives, face-to-face and teleconference interviews were held with the identified stakeholders. The identified stakeholders were all familiar with various aspects of interoperability through experience with open source software development and OKI.

2.2. The Respondents

Over 30 people from 10 organizations were interviewed for about 60 minutes each, either individually or in small groups. Appendix A: Respondents contains a complete list of the respondents.

A conscious effort was made to collect responses from the perspectives of participants in the development of large-scale applications and enterprise systems. The respondents for each organization ranged from CIOs and CTOs, to senior architects and engineers, to Project Managers, to programmers. The table below shows the organizational and project affiliations of the respondents.

Organization(s)	Project(s)/Product(s)
Indiana	Kuali Financials, Sakai
Michigan	Sakai
Cambridge	Sakai
MIT	OCW, DSpace, Stellar, OKI
Stanford	OKI, Sakai, libraries and assessment
UBC	Kuali Student
CSU	Digital Marketplace, MERLOT
PSU, Moodle Rooms, Harvest Road, Angel Learning	LionShare, Moodle, HIVE, Angel Products

Almost all of the respondents had experience directing or participating in enterprise software development projects. All had broad prior experience with the general issues involved in this study, and almost all had some direct experience with the OKI OSIDs in a previous or current project. Some had experience with developing or using other open source collaborative products, e.g., Apache or Tomcat. The respondents had well thought out opinions regarding the benefits of interoperability, the competing objectives in decision-making and development, and the utility of the OSIDs.

2.3. Interview Questions

The respondents were asked which criteria they use to select an approach to integrating system components and how they make trade-offs between providing interoperability and the constraints of operating within a given project's budget, time, and personnel constraints. They were also asked what specific experience they had using the OKI OSIDs.

Q. What is your current position and relevant activity, and what is your general experience?
This question either confirmed or collected information about the perspective of the respondent, his or her role in current projects and the level of experience and expertise that was reflected during the interview.

Q. What do you look for when evaluating approaches, standards and resources for increasing interoperability?
This question elicited criteria by which the utility and applicability of the OSIDs and other design and development tools and methodologies could be evaluated and compared.

Q. How do you balance the cost and benefit of including interoperability as a project objective?
This question brought to light the project parameters that influence the perceived benefit/cost of providing interoperability in a given project.

Q. What kinds of risk influence your choice of approach, and how do you identify and eliminate or mitigate risk?
This question served to identify real and perceived risk of various kinds: scope, schedule, budget, distributed teams, vendor lock, technology change, etc. and elicited a prioritization of risk management or avoidance techniques.

Q. *How did/could the OKI OSIDs and other approaches leading to interoperability demonstrate positive or negative utility in the context of your project?*
If the topic of the OSIDs did not come up spontaneously, this question was asked late in the interviews so that discussion took place in the context provided by responses to the previous questions.

3. Findings

This section presents a synthesis of the answers received in response to the interview questions discussed in Section 2.

One of the primary purposes of this study is to agree on criteria and evidence for making concrete decisions about whether and when to use a particular approach or approaches to provide interoperability in educational enterprises and, specifically, to enable decisions about whether and when the OSIDs have interoperability value in a particular enterprise context. From the respondents' responses and discussion during the interviews, it was clear that not all projects attach the same priority to interoperability. Different approaches fit different project needs, and some project teams are more capable of using one approach over another. In other words, *there is no simple, one-size-fits-all answer to questions about which approach to interoperability maximizes ROI.* Figure 1, below, illustrates some of the key contextual dimensions that must be taken into consideration.

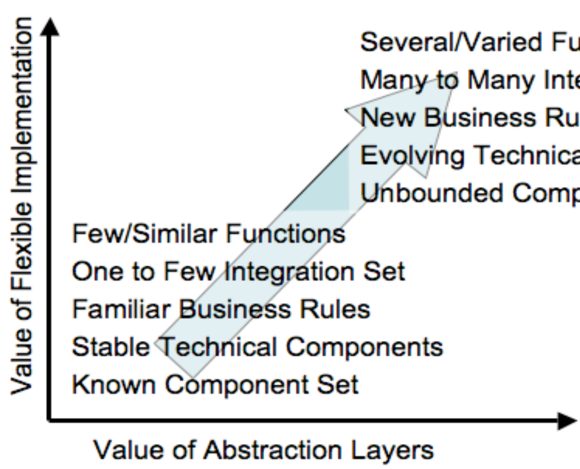

Figure 1. Project requirements impact ROI from flexibility and abstraction in solutions.

Figure 1 illustrates factors that respondents identified as determining the benefits of implementing various interoperability approaches, both architectural and component-based. In general, the greater the variety and change or uncertainty in either requirements or the system's environment, the greater the value that can be returned by insulating parts of the system from each other by means of abstraction layers (e.g., OSID interfaces) or interoperable components (e.g., COM/CORBA objects, Web Services). There are many approaches to accomplish this insulation, and usually there are various ways to implement any particular approach; consequently, this is an area in which attention to terminology is particularly important. For example, an interface definition is an architectural abstraction, while an interface implementation is a functioning piece of code best understood as a piece of software engineering and/or a system component. Conflation of the two ideas can lead to miscommunication and frustration, as (e.g.) developers looking to borrow an interface implementation are confronted instead by an interface abstraction that they cannot use without investing substantial engineering effort—and the concrete benefits of which are impossible to determine *ex ante*.

3.1. Interoperability Criteria

The respondents were probed to identify their approaches and decision processes about implementing interoperability features. In response to the interview questions, respondents noted that:

- Not all projects require external interoperability. Efforts to increase interoperability should be linked to projects for which cross-functional, cross-institutional or cross-vendor integration is a desired stakeholder feature in the project.
- Implementation of the OSIDs, in particular, would best be undertaken in the context of integrating multiple component services in a many-to-many relationship, perhaps even across multiple enterprises.
- Both a comprehensive, consistent model and a sound engineering and development approach contribute to the ROI for interoperability. Approaches should be evaluated in terms of their respective value in the context of a project, not by comparison to each other.
- In a given project context, the value of an interoperability approach may derive from:
  - The particularly effective way it enables functionality
  - The experience gained from its design and implementation
  - Its usefulness as an initial model or example
  - A given project could derive more value from combining several interoperability approaches than from choosing one to the exclusion of others.

The respondents agreed that good interoperability-in-practice results from *both* good design *and* good implementation of a system and its components; neither alone is sufficient. They also said that particular approaches to interoperability generally tend to emphasize either long term ROI (e.g., via an implementation-neutral architecture), or short term ROI (e.g., via implementation-specific engineering decisions), but not both. An architectural point of view will tend to emphasize general-purpose interoperability through high-level abstractions, which requires making an investment of project effort to comply with the abstractions as the software is developed. An implementation-oriented point of view will tend to emphasize the efficiencies of dedicated interfaces and specific implementation paradigms, which requires investing project effort in developing those interfaces. *These distinct foci, coupled with the need to include both good design and good implementation in order to achieve interoperability, suggest that no single interoperability approach is likely to be sufficient for any reasonably complex project*. Instead, the two different points of view will most likely need to be brought together, such as when the architecturally oriented OSIDs are implemented using the production-oriented Web Services paradigm, or when a Web Services implementation complies with an OSID-based design.

This insight is an important reason that questions of the form, "Should Project X use OSIDs or [for example] Web Services?" have no useful answer outside of the context of the requirements and resources of "Project X." This is not to say that the question has no answer or that answering it can be avoided; it is only to say that the question cannot be answered in the abstract, but must first be disciplined by a particular set of facts structured by a particular context. In pursuit of at least a somewhat generalized answer to the question, the respondents identified a number of general criteria for selecting projects that are likely to benefit from higher levels of interoperability and for deciding which approach(es) to use, within those projects, to address interoperability requirements. Those criteria are summarized in the decision matrices in Appendix B: Choosing a Project and Appendix C: Choosing an Approach.

3.1.1. Project Context

Practically every respondent said that design and development decisions around interoperability involve trade-offs between long-term and short-term goals, as well as between stakeholder requests and resource constraints. The respondents identified the following as contextual dimensions that shape interoperability decisions:

- Stakeholder requirements for interoperability
- Budget, schedule and staffing constraints
- Risk
- Production level requirements for:
    - Program performance
    - Security
    - Scale
    - Component choice
    - Type of function

They also agreed that an approach must fit the problem, by taking into account specific project requirements and circumstances.

Several respondents noted that the benefits of different interoperability approaches and resources vary significantly over the life-cycle of the project, as illustrated in Figure 2, below.

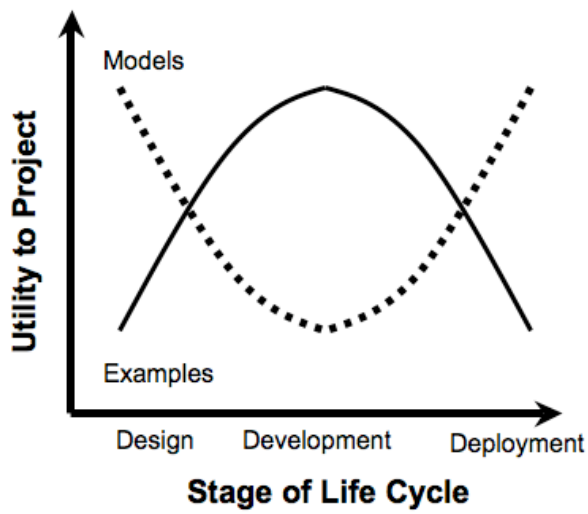

Figure 3. The utility of models and examples changes during a product life-cycle.

Figure 2 illustrates how the respondents described the relative utility of models (architectural abstractions) v. examples (prior implementations) during the life-cycle of a project. Models and examples co-exist in any usable interoperability approach, and usually each can be inferred at least in part from the other.

Models, whether explicit or inferred, have value because they provide a pattern to be imitated or adapted in designing the component service to be implemented. A model that is a recognized standard-in-use has enhanced value because it promises greater return due to its pre-adoption by others. Figure 2 suggests that, once a specific design has been created, the utility of the generic model declines because its influence is already incorporated in the design. On the other hand, the value of the model increases once again, during and after deployment, through its use in documentation and training as well as for post-development integration tasks.

Conversely, Figure 2 suggests that the value of an example is highest during development, when it provides practical guidance concerning specific programming approaches and engineering details. If the example is a reference implementation it has enhanced value because it authoritatively implements the underlying model. On the other hand, an earlier example diminishes in value once the project itself is available; it becomes useful again only in the deployment stage, where it may be useful for integration purposes.

3.1.2. Developer Support

All of the respondents said that project leaders and programmers have to make everyday trade-offs between best practices, schedules, and cost constraints. Those tradeoffs will be wiser and lead to better ROI if there is a substantial body of knowledge and wisdom available for the project teams to exploit. Interoperability approaches that suffer from a lack of these resources operate at a major disadvantage, and may confer even more substantial disadvantages on their adopters.

The project manager and programmer respondents, who are directly responsible for implementation, emphasized that if a particular interoperability approach is mature, it is more likely to be used because developers can find a community of practice, available documentation, examples, and training, and readily available staff.[2] This dynamic is shown graphically in Figure 4.

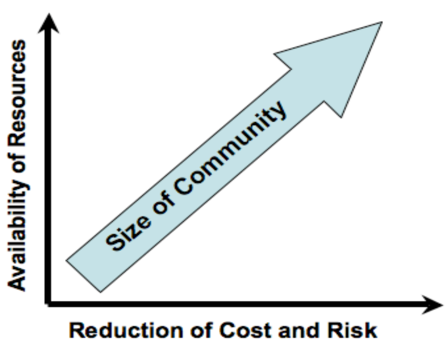

Figure 4. Adoption impacts resource availability, cost, and risk.

[2] Even though respondents routinely described resources in terms of community size, the role of “size” or “maturity” in supporting usability should be interpreted cautiously. More precisely, respondents were discussing the ‘vitality’ or ‘support effectiveness’ of the community—two concepts that correlate with but are not identical to either size or maturity. Over time, even (especially!) large, mature projects can become moribund or irrelevant. A few respondents made this caveat explicitly; it seems likely that most others would agree if asked.

As Figure 3 shows, in general, the larger the supporting community, the more likely it is that a project can draw upon resources such as documentation, programmer tools, practice guides, and prior implementations. A larger community thereby distributes costs and removes or shares risks, which would otherwise have to be borne by the project. New interoperability approaches often lack a community and its resources. In order to compensate for this lack of expertise, other types of support will be required for a successful implementation; examples might include a strong, local leadership commitment to using a new approach, or the hiring of a consultant to fill in gaps in expertise. Intervention can also take the form of a group of early, sophisticated users who are motivated to help each other and the nascent community to "bootstrap" their shared understandings.

3.2. OSIDs as a Means to Achieve Interoperability

The OSID approach to interoperability is concerned with developing abstractions to cope with the many different ways to implement services in-context. In the context of the development of services for enterprise services architectures, for example, there may be a need to facilitate integration between service implementations and different enterprise service bus (ESB) implementations. In such a case, implementing an OSID abstraction might shield a service from knowing whether it is implemented on top of an ESB or via some other approach. Similarly, an OSID implementation might be used to hide implementation details between various layers of the enterprise architecture; for instance, to allow institutions to mix-and-match enterprise middleware products (such as workflow engines, ESBs, and business rules engines) from different sources.

The OSIDs at present show promise for supporting interoperability, but also present some barriers. Our respondents agreed that the appropriate measure of the OSIDs' utility, as well as that of other approaches to interoperability, will be the degree to which using them facilitates the development of interoperable service components and reduces the effort to integrate or replace services in production enterprise environments. The consensus is that some additional work remains to be done before the OSIDs can meet this standard.

For example, both users and critics agreed that, as delivered in the currently available documents, examples, and implementations, the breadth of use cases and thinking that went into defining the OSID interfaces does not provide sufficient support for decision makers and programmers to work on their own. In particular, it was noted that the use of a single, high-level abstraction to cover numerous specific implementations, while it increases the potential power of the OSIDs, also increases the difficulty of translating them into practice and makes the provision of suitable examples and other implementation resources—which are generally lacking at present—even more imperative.

Limited adoption as an obstacle to further adoption is characteristic of the "small community" problem identified in Section 3.1.2, above. For the OSIDs to become more widely useful, in the near term some leadership is needed to coordinate the specialized assets necessary for that small community to grow and produce the resources that will attract a larger, more self-sustaining community.

3.2.1. OSID implementation support

The respondents noted that the community of OSID practitioners is small, compared to similar communities for using standards like JSR-170, SRU/W, OAI-PMH, or Z39.50,[3] there are few OSID reference or example implementations, there is a relatively little OSID documentation or support infrastructure aimed at developers, and only a small number of experts are available to provide advice. Programmers and project managers also indicated that the level of abstraction in the OSIDs is inconsistent with (i.e., much higher than) that of popular programming paradigms and development standards such as JSR-170 or Web Services. The gap between the OSID models and mainstream programmer practices has not yet been bridged. Moreover the OKI group itself has limited human resources available to compensate for the limited use of the OSID approach or for the lack of familiarity with its rationale on the part of the general community[4].

3.2.2. OSID Successes

Respondents also agreed that, although support for implementing the majority of the OSID definitions is limited, there have been some successes; in particular, the Repository and AuthN & AuthZ OSIDs are beginning to enjoy increasing use and success.

There have been several implementations of the Repository OSID, covering a variety of production level repositories; some commercial use has also occurred with Apple, Harvest Road, and Cisco. A community of practice for this OSID is established and growing. The Digital Marketplace project, a funded, enterprise-scale project that requires high volume access to multiple repositories, is making use of the repository OSID.

[3] We do not mean to hold up these other standards as paradigms of perfection; in fact, all of them have struggled with many of the issues that we raise here with regard to the OSIDs. However, those struggles have been mitigated by the engagement of large numbers of contributors, sharing knowledge and working together to develop a community of expertise.

[4] In response to these concerns, MIT, the Institute for eGovernance (Andra Pradesh, India) and CalState have formed an ad hoc consortium to address the need for better support for achieving interoperability using the OSIDs and other means. The goal of the consortium is to create a self sustaining organization that is focused on driving the use and adoption of OKI concepts and methodologies to solve specific problem sets. An example of such a problem-set is the AuthN/AuthZ/IdM need in CalState's Digital MarketPlace. Several organizations already have indicated their intention to participate in the *ad hoc* group. The Open University of Catalunya is in the process of joining the three original founders of the group. Contributions have been received from the first participants. An initial public announcement was made in January 2007. Training sessions were held then, and a workshop/demonstration event is scheduled for August 9-10, 2007. This meeting will take place in conjunction with the MERLOT International Conference, and the program is being planned with MERLOT and IMS.

Similarly, a community of early users of the AuthN and AuthZ OSID definitions is beginning to form. According to several respondents, approaches to identity management seem to be coalescing; however, as yet no standard approach has emerged. The nascent AuthN/AuthZ OSID community is working to make those OSIDs the foundation of such a standard approach. The Digital Marketplace project, which requires an Identity Management solution for ecommerce transactions and for other functions involving trusted consumer and supplier identities, is beginning to use the AuthN and AuthZ OSID definitions in its Oracle infrastructure. The Kuali Student project also has a compelling Identity Management requirement for trustworthy management of student, course, and administrative information. The designers of KS are using the OSID definitions in their planning processes, and discussing their implementation once the build-phase of the project begins.

More generally, particular projects have focused on individual OSIDs or a small cluster of related OSID definitions, rather than on using all of the OSIDs at once. However, several respondents from several organizations said that the overall list of enterprise functions that are provided by the OSIDs, taken as a whole, is well considered and has value in designing implementations. For example, Middlebury College has implemented most of the OSIDs in various production teaching and learning systems on campus, and Moodle Rooms is in the process of implementing a comprehensive hosted LMS service offered at "one dollar per student per year, with no switching costs." This business model is based on "combining OSIDs, IMS specifications, and other standards."

3.3. Other Means for Achieving Interoperability

As mentioned earlier, the OSIDs are not the only means of achieving interoperability. Any interoperable components that are adopted in many projects and are used across a variety of production contexts become *de facto* standards, regardless of whether standard definitions or abstract models of their functions exist. Proven implementations are operational definitions of interoperability and can and do serve additional duty as design models and reference implementations.[5]

There are many standards, or quasi-standards, which correspond to individual OSID definitions. In terms of AuthN/AuthZ alone, for example, Java's JAAS provides an abstraction for AuthN & AuthZ, Sun's Pluggable Authentication Model (PAM) is a widely used AuthN scheme in Linux, Unix, and other applications, and AAI (Authentication and Authorization Infrastructure) from

[5] For example, respondents from Indiana indicated that the Kulai Enterprise Workflow Engine (KEW) has emerged in practice to be interoperable and is evolving toward a standard implementation and model for use outside the context in which it was developed. Respondents from MIT indicated a desire to collaborate in consolidating the utility of the OSID Workflow model and KEW.

the Swiss Education & Research Network, is an AuthN & AuthZ API that is built on top of Shibboleth, but may be expandable to other concrete AuthN & AuthZ instantiations. These efforts match, complement, or in some cases even offer more functionality than the equivalent interface definitions from OKI, but they are effectively standalone, limited to these particular interfaces.

On the other extreme, Web Services standards apply to essentially any interface problem; as such, they can be understood as even more widely applicable as an interoperability approach than are the OSIDs. This makes them a low-cost and particularly effective approach in one-to-one or one-to-many interoperability scenarios, such as Web 2.0 "mashups." However, they achieve this scope by eschewing any domain specificity whatsoever, leaving the entire burden of achieving domain-based interoperability and maintainability to the developer. This makes them potentially a poorly-scaling solution in a many-to-many environment—such as an enterprise environment, where many services need to be coupled in many combinations to support many user-facing applications, all in the same context. In these situations, absent some centrally imposed guidelines or controls, the very flexibility of Web Services implementations can lead to "combinatorial explosions" in the number of cases that a developer or system administrator needs to consider. An important segment of the emerging implementation literature on SOA, for example, is concerned with questions of how to achieve—and especially to maintain—the effective integration of large numbers of Web Services in many-to-many relationships on an enterprise scale; in fact, in a clear sign of the significance of the problem, startup ventures are emerging to offer tools that attempt to minimize the risks and ease the maintenance tasks related to such activities.

Finally, some vendors offer solutions that are more comprehensive than single OSIDs as well as more domain-specific than Web Services. For example, within its Windows development APIs as well as its .NET platform, Microsoft has abstractions for AuthZ & AuthN, Repositories, and Workflow; however, these are tied to specific Microsoft technologies, i.e., Active Directory, SQL Server, etc. Likewise, the Workflow Management Coalition (WfMC) provides a specification and reference model for an interoperable Workflow interface, but it does not provide an abstraction layer to plug in different workflow engines; instead it achieves interoperability through the use of an XML based interchange language, XPDL.

In summary, the OSIDs are generally more open and implementation neutral than the various solutions above. Where they fare less well, by comparison, is in the dimensions discussed throughout Section 3.1: maturity, implementation frequency/community vitality, and developer support. Fortunately, identifying and conducting interoperability experiments with the OSIDs will simultaneously provide information about the prospective value of the OSIDs for other interoperability challenges, and help to develop the other resources needed for easier, wider adoption.

3.4. Potential Test Beds for Interoperability

The Mellon Foundation requested that Ithaka and MIT ask respondents to identify potential test-bed projects that could serve as platforms for the investigation and enhancement of interoperability among Mellon-supported projects more generally. When respondents emphasized the importance of context in assessing the need for interoperability, it became apparent that, in order to identify suitable test-beds, it would first be necessary to identify guidelines or criteria for selecting an interoperability-friendly project.

To be effective, a test-bed environment must permit the assessment of the OSID implementations on several key dimensions. First, it must assess the costs, risks, and stressors involved in implementing OSIDs in a many-to-many environment. This requires, for example, a project that brings together multiple legacy systems interacting with multiple, project-internal services on multiple interface dimensions. Second, it must assess OSID performance "at scale." One of the unanswered questions about the OSID implementations is whether the additional layer of abstraction that they impose will cripple performance in systems that must push many thousands or millions of transactions per day through the OSID interfaces. Third, it must assess the ability of a project not designed specifically for the OSIDs to adapt the OSIDs to a useful purpose. The OSIDs were originally designed as the foundation of a learning management system, but for them to be effective, their abstractions must also make sense in the internal context of the "foreign" systems to which learning management systems connect, such as financial and student systems. To understand where and how the OSIDs may be useful, it is necessary to determine if the OSID abstractions are truly generic only within the context of learning management, or if they generalize broadly to other higher education systems as well.

Finally, of course, the project must be willing and able to implement the OSIDs as an interoperability solution on at least a trial basis, and in a production-level environment. Because of the interview focus on assessing the OSIDs, it was this aspect of the question that received most of the respondents' attention.

3.4.1. Choosing an Interoperability-Friendly Project

The respondents identified several dimensions of enterprise project environments that benefit from or require interoperability and are therefore more likely to demand implementations that will improve the level of interoperability. Appendix B: Choosing a Project, lists the criteria comprehensively and illustrates their applicability. The basic probes/dimensions identified by the respondents were:

- What production level project is underway or planned?
- What is the project requirement for cross-functional, cross-application, and/or cross-institutional interoperability?
- What time, money and staff are allocated?
- What specific metrics for cost/benefit are identified?
- What interoperable implementations are in use?
- What model or standard applies?
- Is support for developers and decision-makers available?

Answers to these questions will provide data with which to assess the likelihood that a project has the need and the wherewithal to pursue interoperability seriously as a deliverable.

Several more specific decision criteria also emerged from the interview process for selecting test-bed project environments and development approaches that would tend to maximize the benefit/cost ratio of interoperability. These include:

- A production project with a defined budget and schedule

- A project with specific, high-priority requirements for interoperability
- Compelling interoperability use-cases with quantifiable metrics of both cost and benefit

Respondents also suggested that an ideal test-bed environment would possess

- Prior implementations and existing models that use the approach to be taken, both to reduce the difficulty of the task and to relate the findings to other empirical evidence
- A community of practice that is committed to active support for the approach, and/or
- Specific budget, time and dedicated support to compensate for absence of prior work or lack of community

3.4.2. Identified Potential Test Bed Projects

The respondents identified two enterprise projects as exhibiting many of the features required for successful interoperability implementations and as high-value potential testbeds for an evaluation of the OSIDs:

- The CSU Digital Marketplace (DM) project is an ambitious, system-wide project to integrate the market for learning objects among commercial and other producers, faculty (who identify materials as part of course curricula, and thereby serve as unpaid 'brokers' for the materials to the students) and student consumers. When complete, DM will coordinate the activities of tens of campuses, hundreds of integrated software systems such as campus-based LMSs and financial systems, hundreds of vendors, tens of thousands of faculty, hundreds of thousands of students, and potentially millions of transactions a day at peak periods—almost all of which are in many-to-many relationships. As a system that intersects the financial and teaching/learning domains, it can potentially make use of a broad spectrum of the OSIDs, but it is also different enough from the implicit use-cases of the OKI project that it poses some interesting implementation challenges as well.
- The Kuali Student (KS) project aims to produce a next-generation student services system that will possess some revolutionary features. As the "mission critical" application for any college, a student system touches nearly every aspect of the enterprise, including teaching/learning (through its Registrar functionality). It therefore can make use of the OSIDs, and like DM, it does so from a perspective outside of the one in which the OSIDs were created. Because it coordinates so much of the academic enterprise, and because its architecture emphasizes a 'pure' SOA approach with minimal redundancy across services, KS is rich in many-to-many services relationships. The production level of a KS implementation will depend directly on the size and activity of the institution, so while the absolute scale of the resulting systems is not yet known, they will clearly operate at a scale that is as great as the scale of any other enterprise system that the same institutions are likely to run. Core partners in the KS project include state systems of higher education and other very large users, so the scale achieved in testing will serve as a reasonable predictor for virtually any other higher education institution considering the use of the OSIDs.

From a variety of additional perspectives as well, these projects provide an excellent context for testing different approaches to enterprise interoperability and developing systematic decision procedures for evaluating and deciding how, when and where to use a particular interoperability approach. Both are enterprise scale projects with long life expectancies; both also have challenging requirements for performance, security, availability, redundancy, and life-time maintainability. All of these are features and characteristics that potentially bear on the short and long-term ROI of the use of OSIDs and/or the ROI of interoperability generally. Both involve multi-stakeholder collaborations to produce a design and implementations that will be used in a variety of organizational environments. Both have some commitment to exploring multiple interoperability approaches; KS, for example, plans to use Web Services as well as (possibly) the OSIDs. Better still, some functions required by the projects overlap; for instance, identity management and grade recording are common requirements. Finally, the timeframes of the projects are sufficiently compatible that each can benefit from collaboration with the other.

4. Areas of Focus for Success

Assuming that either the DM project or KS, or both, are available for OSID testing, further discussion among technology providers and stakeholders in the DM project and the KS project environments is required in order to develop detailed interoperability requirements and to make a decision about which specific approaches to use and how to use them. These conversations were started at the March workshop, to which representatives from both projects were invited. In preparation for those talks, discussions with several respondents identified specific topics that are ripe for further work; that is, areas in which demand for interoperability exists, and in which existing OSID models and implementations of interfaces and components seem ready to be applied in production systems.

The following sub-sections each detail an area of focus that emerged from the interviews and that, if pursued, could result in increased interoperability and greater integration of components across many educational enterprises. The descriptions below consolidate suggestions and comments that were made by various respondents.

4.1. Identity Management

In a world of escalating privacy and security concerns, identity management (IdM) is a high-priority task. In the world of higher education, IdM must be implemented with an eye to interoperability, because of the inherently cross-institutional, collaborative nature of the scholarly enterprise. Models of effective, cross-domain IdM are coalescing, and implementations are being developed by institutions, commercial vendors (IdM suites are available from several major vendors), and collaboratively (e.g., the open-source CAS project).

One such model and implementation, using OSIDs for AuthN and AuthZ, is already being developed for the DM project by CSU and OKI. IdM is a many-to-many problem for DM because the project must be implemented at more than thirty campuses, each of which has its own legacy IdM regimes (there is some duplication, of course), ranging from commercial systems, to open-source solutions, to campuses where the 'system' is as yet no system at all. It also must integrate IdM for the content vendors. Those campus regimes must be connected via DM to the many legacy systems (e.g., the various campus LMSs) into which purchased content must flow. The KS project planners are also examining the OSID approach to AuthN & AuthZ, as well as existing OSID definitions and interaction specifications. A collaboration between the two projects on IdM-related interoperability would have the potential for significant impact on other, future projects, by creating highly visible and re-usable models for single and multi-institutional environments.

4.2. Repository Submission and Federated Search

Scholarly and instructional content in higher education comes from many sources, e.g., commercial publishers, Open Courseware, and individual or collaborative scholars, and flows into many sinks, e.g., repositories such as Fedora, D-Space, or Harvest Road, learning management systems such as Sakai, Moodle, and Blackboard, content management systems such as Plone and iTunes and DIVA, and other, more ‘lightweight’ stores such as faculty directories and web sites. It is also accessed from many points: user applications for submitting and searching content are implemented in such projects as MERLOT, Connexions, SCORE, VUE, etc. This sort of many-to-many world is an ideal evaluative space for the OSIDs. An infrastructure that will support high volume submission, search, retrieval, and e-commerce across a variety of commercial and open source repositories is required and is being developed by the DM project. Institutions are likely to adopt the resulting model and implementations as a standard implementation because of the comprehensive vendor participation by the DM project. Future developers will be able to gain interoperability with almost all major higher education content sources by adopting the DM project’s reference implementation.

Various respondents noted that repository and search is a particularly appealing topic area for OSID investigation because it tests the OSIDs where they are strongest: the repository OSIDs, along with supporting metadata standards and component services for archiving, accessing, and using digital content, have seen a great deal of development effort over the past several years. Consequently, many parts of the DM project’s end-user requirements for creating, managing and using content from multiple sources will find ample supporting resources in the OSID community. Additionally, several respondents opined that now is the time for an ambitious integration of the repository OSID into DSpace or other repository implementations, and into the MERLOT/GLOBE user environment and various approaches to federated search.

4.3. Workflow

Auditable management of workflow is required in the DM project for e-commerce transactions and for tracking compliance with accessibility regulations; of course, many other educational functions, such as course management, scholarly collaboration, and grade recording, benefit from workflow management, too. The Mellon-funded Enterprise Service Bus project, which is designed to define and determine the requirements for ESBs in higher education enterprise architectures, could develop a model and reference implementation for implementing workflow across both academic and administrative functions, perhaps by leveraging existing open-source workflow engines, e.g., Kuali Enterprise Workflow, to provide re-usable implementations. Virtually all higher education institutions of any size use some type of workflow engine today, somewhere, but few if any are able to provide integrated, interoperable, enterprise-wide workflow functionality at present—and even fewer can accommodate workflow interoperability across institutional boundaries. A production grade, re-usable implementation of highly interoperable workflow has great potential for widespread re-use across educational enterprises.

OKI has a Workflow OSID, which could be adapted to support the re-use of the Kuali Enterprise Workflow engine by the DM project and KS as well as the ESB project. Respondents pointed out well-known and supported integration points for individual assessment systems, grade books, grade recording services, and student information systems. They also suggested that an interoperable workflow engine would encourage a marketplace of interoperable tools and services to emerge, which would ideally offer greater choice of component services, and greater benefit to many projects. The Indiana respondents stated strong belief in the interoperability and value of KEW, and indicated a commitment to supporting its use for a broad range of workflow applications by other organizations. A specific solution to the workflow requirement within the DM project could be to integrate KEW in the context of efforts that are underway to identify, validate, and develop a standard integration approach, e.g. the Grading OSID, the Sakai Grade Book project, IMS QTI and TIF, or the Workflow OSID. The DM project and KS pointed out that these project environments would also provide a useful range of workflow applications for testing, such as course registration, a review or audit of compliance with accessibility regulations by the CSU accessibility office, and clearing e-commerce transactions.

Use of the Workflow OSIDs has the potential to allow application designers to choose a workflow engine which would fit their needs, without the burden of designing with change in mind, i.e., changing the workflow engine in the future if needs require it.

4.4. Interoperable Gradebook

Grade recording is essential for the teaching and credentialing of students. The DM project respondents also noted that closing the loop between learning experiences and outcomes is essential to providing accountability for institutional, faculty, and student performance, as well as for interventions and support provided by business model changes, student services and faculty development. Commercial, open source, and custom LMSs commonly include a grade-book function that captures and presents student assignment outcomes, with aggregations for specified periods. These solutions are aimed primarily at faculty users, and may not be considered authoritative; for institutional purposes, they interact with a different system-of--record, maintained by the Registrar.

Respondents from the DM project, KS, Stanford, Sakai, Angel Learning, and Moodle Rooms all pointed out the opportunity to consolidate local and/or application specific implementations for integrating assessment engines, LMSs, and SIS systems. Both DM and Kuali Student require grade recording capabilities; ideally, those capabilities should be built for re-use and connected to other system functionality via a workflow interface. Such a gradebook could also be useful as a replacement/upgrade for the less-interoperable versions presently found in Sakai, Moodle, and proprietary LMS products, and may be useful for other student information systems as well.

Development of a reusable, interoperable gradebook can benefit from several previous standards-setting efforts: IMS Enterprise specification, IMS Question and Test Interoperability specification (QTI), IMS Tool Interoperability Framework (TI), and the JISC Tool Interoperability Project. Indiana, Michigan, and Moodle Rooms respondents pointed out that a demonstration of a re-usable grade-recording module using some of those standards has already occurred at the IMS alt-i-lab 2005.

4.5. Time Period Modeling

One additional area that was identified by respondents is time period modeling. KS respondents stated that KS and the DM project would need abstract time periods, which could be customized for local use, and be consistent with existing transcript standards, license terms, and academic processes. The KS project will be working on a suitable implementation in this area, and there is a potential for an interoperable standard to emerge; however, because the project was of interest primarily to those two institutions, and there is not enough information available to assess the likelihood that their efforts could result in a broader standard (especially given the universality of time-management as an IT concern, and the highly volatile and mostly ineffectual nature of time and calendaring standards efforts to date), there is no further discussion of this area in this report.

5. Conclusions Reached

Based on the interview responses, approaches to providing interoperability develop gradually along a continuum from "Hello World" demonstrations of capability, to "Real World" implementations of functionality, to "World Wide" adoption and use. The metrics and expectations for ROI increase in detail and specificity at each successive stage. The respondents in this study have clearly moved into the "Real World" phase: they now expect quantifiable estimates of integration cost and system performance and preliminary metrics to support process control and risk mitigation. They also expect that accumulated practical experience will result in reliable predictions for life cycle longevity, the ability to evolve, business results, etc. Several respondents pointed out that the early claims for benefits and outcomes from using the OSIDs could not be substantiated because they were based on Hello World implementations. Expressions such as "look for a crowd" and "[look for] at least 12 FTE's involved in support" indicate the respondents' current expectations for mature approaches. While there is significant desire among many respondents to use World Wide standards for interoperability—a desire seen most clearly in the widespread affection for Web Services—there is also the realistic sense among respondents that solutions of that scope are not yet commonplace, and that for many purposes, Real World is adequate and cost-effective.

Apart from their Hello World status, respondents provided several reasons for the current lack of interoperability and lack of adoption of the OSIDs. Interestingly, most of their explanations touch in some way on the many-to-many benefits of the OSIDs, and the ways in which those benefits are difficult to realize in real-world projects:

- The OSIDs support long term interoperability on an enterprise scale, requiring a commensurate level of effort to implement; their value-proposition for one-to-one or one-to-many interoperability needs is usually less than the value proposition of at least some competing approaches, and necessarily less than their value for many-to-many interoperability challenges.
- Projects prioritize interoperability as one of many tradeoffs; others include functional requirements, project constraints, and implementation risks. In making those tradeoffs in previous projects, project managers and developers generally have not found many-to-many interoperability to be a high enough priority to make the value-proposition of the OSIDs compelling.
- There is a 'first-mover disadvantage' (a.k.a. a "chicken-and-egg problem") in adopting a many-to-many interoperability standard. In a many-to-many environment, the first few projects to implement interoperability bear disproportionate costs and realize few immediate benefits (because they break the ground but have few partners as yet to help generate returns), while large benefits and minimal costs accrue to later adopters (who can adapt the work of the early adopters to cut their own implementation costs, and who arrive in an ecosystem already populated with many partners from which to generate returns). This creates perverse incentives against early adoption—especially in an environment where participants are seeking competitive advantage through interoperability—unless special efforts at coordination are made.
- The art of managing many-to-many interoperability is neither intuitive to most developers, nor widely familiar, which makes the costs and risks of using the

> OSIDs higher, other things equal, than the costs of learning and using one-to-one or one-to-many competitors. Moreover, those costs have been further inflated by the lack of training and development resources. The availability of tools, examples, guidelines for practice, staff skills/experience, and community support all influence choices between interoperability approaches, as well as the amount of "extra" effort that is devoted to increasing interoperability.

The patterns in these responses suggest that systematic support for decision makers, project managers and programmers is necessary to close the gap between OSID-based interoperability as an abstract concept and interoperability as a delivered feature. Pragmatic objectives, archetypical examples, explanations that lead from examples to generalizations, and insightful and consistent models, are necessary for a community of practice to develop. Some additional, outside support to projects contemplating use of the OSIDs may be required to overcome the first-mover disadvantage.

These general conclusions should be tempered by context and lessons learned. In the following sub-sections, the authors offer some advice to anyone attempting to set expectations for ROI from interoperability in a specific project context.

5.1. Project goals and project constraints determine the value of interoperability.

Many respondents commented that interoperability never will be complete or perfect, nor will most sensible project leads ever invest significant project effort to provide a benefit outside the project scope. A project's requirements and its practical constraints determine the crossover point where cross-functional and/or cross-institutional, cross-vendor interoperability provides a net benefit in return for extra effort in the project at hand. We have shown this graphically in Figure 6.

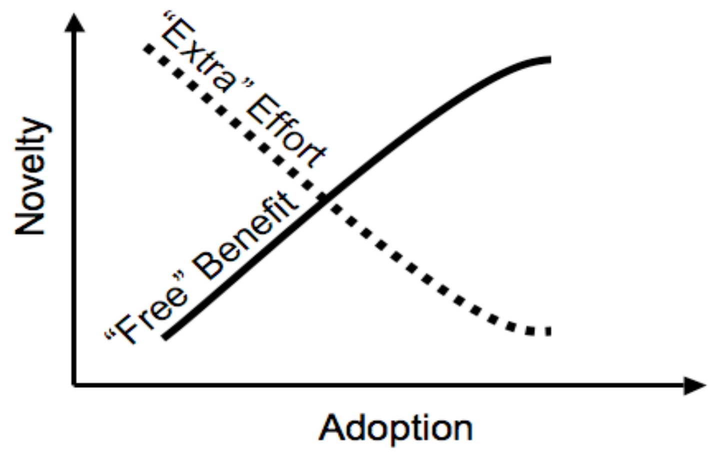

Figure 6. Novelty and adoption in determining the effort and benefit of implementing interoperability.

The above figure illustrates an observation made by several of the respondents: any relatively novel approach such as the OSIDs or other abstractions, requires extra effort. Whether it is perceived or real, "extra" effort is difficult for a project manager or programmer to justify. Early adopters are invariably motivated either by a specific requirement or by some objective other than (narrowly defined) project efficiency.

Figure 4 suggests, and the respondents confirm, that projects where interoperability is an intrinsic production level requirement would have the best prospect of improving interoperability. There is little prospect that external interoperability will be implemented unless it is a project requirement. Not all projects, though, justify the effort to provide external interoperability.

Respondents also indicated that outside incentives could shift the crossover point in Figure 4 by increasing the benefits of interoperability to the project. To be effective, though, such interventions must also hold realistic expectations regarding the project at hand. For instance, no amount of extra funding is likely to move a project to develop a secondary feature such as interoperability if it is having trouble meeting crucial deadlines for core functionality. Similarly, projects that have every intention of supporting interoperability, but lack developers qualified to do so, are unlikely to produce satisfactory interoperability results. In other words, firm project requirements, core deliverables, and staffing realities will almost always take priority over abstract goals that are not bound to a specific deliverable. Consequently, it is important to bind interoperability goals tightly and unambiguously to project deliverables, and to ensure that project resources are adequate to accommodate the additional burden.

5.2. Interoperability requires systematic decision-making

An old project management aphorism holds that "a failure to plan is a plan to fail." Respondents echoed this theme repeatedly, stressing the importance of intentional decision-making around interoperability both prior to and during development. The respondents from Indiana University addressed this issue in the greatest detail, describing a well-articulated decision-making process in their institution that addresses many of the project-planning trade-offs described by other respondents. This process balances the benefits and costs of development by the careful prioritization of technical and functional requirements. It also provides a cost and schedule framework for decision-making within and across development cycles for a project. Their two stage estimating and prioritizing process systematically serves to identify tradeoffs and resolve conflicts between short-term time and effort and long-term ROI. Project planning methodologies of this sort are extremely valuable in protecting values like interoperability from the counter-pressures of competing values in a development project.

It is worth noting here, again, the extent to which ambiguous terminology and narrow thinking can also corrupt decision-making. In order to implement "interoperability" in a particular project, for a particular function, it is extremely helpful if everyone involved shares the same understanding of what that word means. It is also unrealistic, perhaps even unfair, to expect a developer who is heads-down on a coding deadline to keep interoperability firmly at the forefront of his or her thoughts; that is properly a job for the project lead and manager.

Using criteria such as those listed in Appendix C: Choosing an Approach, in a systematic decision process like that described above, using unambiguous, agreed-upon language, will help to reduce the threats to interoperability posed by *ad hoc* decision making and ambiguity.

5.3. Uncertainty cannot be eliminated

The variety of institutions and organizations in which the respondents work highlights the extent to which the heterogeneous, decentralized, and individualistic higher education context places high value on interoperability and systems integration. Higher education is also notorious for having ill-defined and ever-changing end-user and business process requirements; consequently, both project requirements and the details of solutions are likely to change as work proceeds. In such an environment, evolutionary software development models can be used to reduce risk and validate design decisions by means of incremental delivery and testing. Their risk-abatement benefits are likely to outweigh their costs, especially as project complexity increases and interoperability becomes more critical.

Returning to an earlier observation about the tradeoff between design and implementation, it seems that higher education as a software development environment rewards projects that integrate both design-time adoption of assets like the OSIDs, which can forestall a great deal of re-design later at the cost of some up-front effort, and code-first solutions, for aspects of the project where the particular "shape of the landscape" is likely to change unpredictably or product lifetimes are likely to be short. Expending design-time effort trying to anticipate the unknowable or the transient is no way to increase ROI; on the other hand, failing to anticipate the knowable or to build in change-proofing is just as wasteful. Picking the right balance will remain, for the foreseeable future, an art in which intelligence will need to be guided by experience.

5.4. Tools and staff skills/experience determine how much interoperability is achieved

Because our respondents came from many levels and roles in many different software development projects, we were able to see that different groups require different types of support in understanding interoperability requirements and formulating approaches. At the same time, however, *everyone* involved in such projects—across as well as within projects—would benefit from a documented set of mutually understandable terms, concepts, policies, and procedures.

Some groups, especially developers, prefer bottom-up scaffolding and support. They seek specialized development tools, training in concepts, catalogs of best practices, libraries of examples, and illustrative guidelines that provide accessible explanations of design abstractions. These tools make it easier for programmers and designers to communicate and create interoperable solutions. On the other hand, other groups, especially architects and project managers, prefer top-down scaffolding, in the form of business oriented examples, economic models, or functionally oriented descriptions of technology. They use these resources to relate project-internal utility more effectively to project requirements and deliverables.

A wise project manager, who is interested in improving interoperability, will ensure that the project has access to both types of scaffolding. A wise leader of an interoperability community, who is interested in maximal adoption of the community's approach, will ensure that both types of scaffolding are available, as richly as possible, to all who seek them.

5.5. Final words

> Preaching won't change practice
> A system is only as interoperable as its least interoperable part.

Even small decisions by individual developers, whether intentional or inadvertent, can have adverse effects on interoperability. If interoperability is to be accomplished, then education and evangelism need to be supplemented with concrete interoperability goals and objectives, integrated interoperability tests, persistent management support for interoperability implementation, and in-project incentives to encourage participants and stakeholders to build in interoperability at every point in the project.

Appendix A: Respondents

Last Name	First Name	Affiliation	Title
Brooks	Lois	Stanford University	Director, Academic Computing
Counterman	Craig	MIT Stellar	Senior Manager, AMPS
d'Oliveira	Cecilia	MIT OCW	Technical Director OCW
D'Souza	Wilson	MIT IS&T	Director Infra SW Devel & Arch
Diggory	Mark	MIT DSpace	Systems Manager
Fairlie	Cath	University of British Columbia	Program Director, Community Source Student Services System
Fernig	Leo	University of British Columbia	Project Manager / Technical Architect
Golden	Glenn	University of Michigan	Software Architect
Gollub	Rachel	Stanford University	Manager, Stanford Digital Repository
Grochow	Jerry	MIT	VP of Information Svcs
Halm	Mike	Penn State University	Special Projects, Teaching and Learning with Technology
Hanley	Gerry	CSU	Sr. Director Academic Technology
Haeusser	Jens	University of British Columbia	Directory, Strategy Department of Information Technology
Henderson	Ray	Angel Learning	VP Chief Products Officer
Hoebelheinrich	Nancy	Stanford University Libraries	Metadata Coordinator
Hyde	Ben	MIT DSpace	SIMILE Project Manager
Iannuzzo	Catherine	MIT IS&T	Infrastructure Developer
Johnson	Nate	Indiana University	Systems Integration Team
Li	Lydia	Stanford	Academic Computing
Mackie	Chris	Andrew W. Mellon Foundation	Associate Program Officer
Margulies	Ann	MIT OCW	Executive Director OCW
McGough	Brian	Indiana University	Kuali Lead architect
Merriman	Jeff	MIT OKI	Associate Director, Office of Educational Innovation and Technology
Mills	Dave	Angel Learning	VP Chief Technology Officer
Norman	John	University of Cambridge	Director of CARET
Posten-Day	Jan	Blackboard	Director, Standards and Interoperability
Reilly	William	MIT DSpace	Technology Projects Manager

Sady	Bryce	BFW Publishing Group	Project Manager
Severance	Chuck	University of Michigan	Software Architect
Sim	Stuart	Moodle Rooms	CTO and Chief Architect
Smith	MacKenzie	MIT DSpace	Assoc Director for Tech MIT Libraries
Smythe	Colin	IMS	
Speelmon	Lance	Indiana University	Course Management Systems
Stone	Larry	MIT Dspace	Programmer
Thorne	Scott	MIT IS&T, OKI	Data Architect
Uyeda	Gord	University of British Columbia	Project Manager / Technical Architect
Walsh	Barry	Indiana University	Director Information Systems
Ward	George	CSU	Technical Lead Digital Marketplace
Wheeler	Brad	Indiana University	VP Information Technology
Wolfe	Robert	MIT Libraries	Metadata Specialist
Wright	Norm	MIT IS&T	MIT Affiliate

Appendix B: Choosing a Project

This decision matrix consolidates criteria from the interviews conducted in this study. To use the matrix, answer the questions in the left-hand column in detail for the project under consideration. Some preliminary details for projects in the areas recommended in the study are included to illustrate the process. Projects with more specific data in more cells are likely to produce more interoperability and more return from effort devoted to providing interoperability.

Criteria	Candidate Projects				
	Identity Management	*Repositories*	*Workflow*	*Grade Recording*	*Other Interoperability features*
What is the project requirement for cross-functional, cross-institutional interoperability?	Most interactions require user identity and/or authorization	Many interactions involve content access	Many functions require workflow manageme nt	Cross-functional requirement to connect LMS, Assessment, SIS; cross-institutional requirement for reporting, transcripts	
What production level project is underway or planned?	DM, KS	DM	KF, DM, KS	LMS vendors, KS	
What time, money, and staff are allocated?	TBD	TBD	May be available	Available	
What specific metrics for cost and benefit are identified?	$, Time quantities, Benchmark function	$, Time quantities, Benchmark function	TBD	$, Time quantities, Benchmark function	

What interoperable implementations are in use?	None	Where? How?	Where? How?	IMS TIF, JISC TIP and product hacks	
What model or standard applies?	Name(s)	Name(s)	TBD	TBD	
Is support for developers and decision-makers available?	Yes – for design	Yes – for design and develop-ment	Yes - for develop-ment	Yes – for design, development and deployment	

Appendix C: Choosing an Approach

This decision matrix consolidates criteria from the interviews conducted in this study. To use the matrix, list the assets and liabilities implied by answers to the questions in the left-hand column for the approach under consideration. Examples of the kind of assets and liabilities that respondents identified in the study are included to illustrate the process. Approaches with more assets and fewer liabilities that apply to the project at hand are likely to produce more interoperability and more return from effort devoted to providing interoperability.

	Assets	**Liabilities**	**Rate Your Approaches**
What kind of community of users or early testers and adopters of solutions is there?	Tools, staff, etc. Multiple suppliers Examples, reference models, etc. exist for estimating and implementation Cost, risk, R&D, etc are distributed. Known unknowns	Customized fit requires work-around, less control over standard May lag tech or business changes	
How well does this use or extend existing practice?	Builds on legacy systems and existing staff skills Predictable change trajectory, available staff	Delays change, Persists risks or inefficiencies	
How well do the models and the tools match the problem and the project?	Easier communication Stronger consensus Leverage of prior work	Big risks aren’t managed, Big mistakes surface late	
To what extent is this a development requirement?	Lowers cost, meets schedule, uses staff, satisfies functional requirement	Sacrifices other benefits or increases other costs	

To what extent is this a Use Case requirement?	Provides high priority end-user function or enterprise capability requirement	Sacrifices other benefits or increases other costs	
To what extent is this a business requirement?	Provides necessary control over IP, price, distribution, partners, contract terms, etc.	Sacrifice other benefits or increases other costs	

Appendix D: March 28, 2007 Meeting

Chair:
Stephen Lucas

Attendees:
Alan Brenner, Lois Brooks, Tom Coppeto, Chas DiFatta, Ira Fuchs, Aaron Godert, Jens Haeusser, Gerry Hanley, Jeff Kahn, Keith Kiser, Steve Lucas, Chris Mackie, Brian McGough, Jeff Merriman, John Norman, Chuck Severance, MacKenzie Smith, Lance Spellmon, Scott Thome, Ed Walker, Barry Walsh, George Ward, Don Waters, Brad Wheeler, and Bill Ying

Meeting Notes

After introductory comments and introductions by Steve Lucas, Ed Walker outlined the meeting objectives and reviewed the agenda. He and Jeff Kahn described the study they conducted, and reviewed its findings and recommendations. From the study, two projects were recommended as context for further work: the CSU Digital Marketplace and the UBC Kuali Student and four project areas were recommended for interoperability projects: Identity Management, Repositories, Workflow, and Grade Reporting.
Gerry Hanley presented an overview of the Digital Marketplace and Jens Haeusser presented an overview of the Kuali Student.
Ed Walker facilitated a general discussion of next steps: goals, roles, concerns, and follow up actions on the four project areas using the "Choose a Project" decision matrix from the pre-meeting report, listed in Appendix B.

Please see http://tid.ithaka.org/oki-interoperability for additional notes and observations from the meeting.