Abstract: SCORM has evolved over the last four-plus years to include a variety of features and capabilities. Certain capabilities are missing, outstanding technical issues remain, and there is a collection of existing and emerging learning technology specifications and standards that are candidates for inclusion in future versions of SCORM. A primary question is what to include, and when, in future versions of SCORM 1.x.

A brief history of the evolution of SCORM is followed by a more detailed presentation of the architectural evolution of SCORM. In turn, this is followed by an outline of current issues that leads to a description of potential standards and features for inclusion in the SCORM bookshelf and SCORM environment.

Disclaimer: The contents of this report are based on research conducted by the Learning Systems Architecture Lab at Carnegie Mellon and sponsored by Carnegie Mellon University and other organizations. The views, opinions, findings, conclusions and recommendations contained herein are those of the authors and should not be interpreted as necessarily representing policies or endorsements, either express or implied, of Carnegie Mellon University, any project sponsor or the Advanced Distributed Learning Initiative.

Contents

Part I -- SCORM Evolution

- Introduction
- A Historic View of SCORM Evolution
- A Technical View of SCORM Evolution
- Generalized Learning Management System Architecture
- Model of SCORM Evolution
  - Learning System Evolution
  - SCORM System Evolution
  - SCORM Behavioral System Evolution

Part II -- SCORM Issues
SCORM 1.x Issues

- Technical Extensions to SCORM
  - Presentation
  - Navigation
  - Assessments
  - Simulation Content
  - Repositories
  - Data Models
  - Learner Profiles
  - Security
  - Communication Models
  - Web Services
  - Collaboration
  - Service-Based Content
  - Localization and Internationalization
  - Accessibility
  - Content Variants
  - Digital Rights Management

- Learning Extensions to SCORM
  - Competency-Based Learning
  - Performance Support
  - Mobile Devices and Mobile Learning
  - Adaptive and Model-Based Systems
  - Intelligent Tutoring Systems
  - Simulation Systems
  - Gaming Systems
  - Enterprise Integration

- Infrastructure Extensions to SCORM and SCORM Bookshelf Management
  - Semantic Web
  - Versioning and Version Management
  - Migration and Backward Compatibility
  - IMS Specification Harmonization
  - Emerging Specifications
  - Specifications into Standards

Part III -- SCORM Futures

- The SCORM Bookshelf
  - SCORM Standards Bookshelf
  - SCORM Environment Bookshelf
  - SCORM Content and Application Bookshelf
  - SCORM Core Infrastructure Bookshelf
  - The Bookshelf Stack

- Criteria for Inclusion of Components in the SCORM Bookshelf
- A SCORM Roadmap
- SCORM Issue Summary
Part I -- SCORM Evolution

Introduction

SCORM (the Sharable Content Object Reference Model), an application profile for interoperability of learning technology systems, is based on a collection of underlying learning technology specifications and standards (the SCORM bookshelf). SCORM has progressively evolved since its initial release in 1999. It currently has widespread support and adoption across all sectors (primary and secondary education, higher education, training, performance support) and diverse communities and industry segments (e.g., military, government, education, medicine, health care, aviation, computing, telecommunications) throughout the world. SCORM has evolved far beyond its roots in the US and its original government and military target audiences.

It is recognized that SCORM is not universal and not all encompassing, but is a starting point for creating interoperable learning technology systems (primarily content delivery systems) and content. Additional features and capabilities can be added to meet the needs of the community, as stated and driven by the community. It is also recognized that due to its historic underpinnings, the architecture underlying the current version of SCORM may not have the flexibility needed to move beyond its roots in support of procedural learning.

This history and background notwithstanding, there is great potential for enhancements and extensions to SCORM. These must be founded on an architectural approach, supported by research, practice and user requirements. Working from an evolutionary background, this report outlines a detailed model for the potential evolution of the SCORM 1.x architecture and illustrates where new features and the supporting learning technology standards could be added to SCORM.

Additionally, this report only outlines the potential technical evolution of SCORM. It does not address the broader issues needed to further the creation of a sustained community of practice surrounding SCORM.

Beyond the architectural evolution of SCORM 1.x, the report includes a discussion of the overall SCORM bookshelf, and an outline of key issues in moving to the next versions of SCORM 1.x.

Any evolution of SCORM 1.x and release of future versions must be carefully planned with considerations of how this will be perceived by the broad worldwide community. Simply adding features because they can be added may not be in the best interest of the community. Given the level of adoption and interest in SCORM, its evolution must be carefully planned and orchestrated. There remains significant danger that changes made too rapidly, or changes not rooted in the community, will either disenfranchise users, be viewed as being driven by a hidden agenda, or may result in community sectors and vendors waiting for SCORM to be finished before moving forward.

A Historic View of SCORM Evolution

The diagram (derived from Computer Based Instruction, Chapter 16, by Gibbons & Fairweather, within Training & Retraining, 2000, Tobias & Fletcher) illustrates the evolutionary history of learning technology systems over the last five decades. It shows two divergent paths.
On the bottom of the diagram is the evolution of procedural types of learning technology systems, typified by computer-based training. This path has led to the types of commercial systems and interoperability specifications, such as the AICC CMI model, that led to the development of SCORM.

On the top of the diagram is the evolution of adaptive and generative systems, typified by AI and model-based intelligent tutors. While some of these systems have moved from research to commercial applications, they have not been widely deployed, and interoperability specifications surrounding such systems have yet to be developed.

The diagram illustrates the eventual need for reconvergence, i.e., providing support and interoperability for advanced adaptive and generative systems in all types of learning systems. As illustrated, there remains the commercialization and additional research activities needed before the reconvergence can begin. While future convergence is critical, there still remains significant work that can be done to further current practice in the more traditional types of procedural systems. Thus, this report deals with how to extend the current SCORM 1.x family of models, rather than addressing the reconvergence and what could be a SCORM 2.x.

As noted above, given the potential downside and misconceptions about SCORM evolution in the broad community, the public position on the evolution of SCORM needs to focus on the SCORM 1.x path. There is still significant life left in the SCORM 1.x family, and insufficient basis to embark on SCORM 2.x.
A Technical View of SCORM Evolution

The diagram illustrates how SCORM, as the bookshelf of standards, has evolved through the inclusion of developing standards.

Initial contributions to SCORM came from the various components of the AICC CMI specification, with the addition of metadata (from IMS and Ariadne) to describe learning materials. In the initial release, SCORM focused on the interchange and interoperability of courses. Also, due to the lack of an available specification to draw upon, SCORM included its own course/content model.

The initial SCORM 1.0 release was followed by the first maintenance release (SCORM 1.1) that also eliminated features derived from the AICC CMI model that were not widely used and that were to be replaced by improved alternative approaches as SCORM evolved. The removal of features in going from SCORM 1.0 to SCORM 1.1 was somewhat controversial, as there was much confusion over the implications of removing features and an incomplete understanding of how and where the features targeted for removal were being used. This clearly indicates that the impact of SCORM evolution on the community must carefully be considered.

SCORM 1.2 provided the first major technical change to SCORM with the inclusion of IMS Content Packaging, and provides the basis for most current implementations and content. SCORM 1.3 follows the trend of inclusion of new components and features through the addition of new specifications to the SCORM bookshelf; in this case, the addition of the IMS Simple Sequencing specification.

The approach for the inclusion of components into SCORM is to adopt existing specifications as the basis for the reference model. SCORM adds both the critical profiling of the specification for the community and the description of how the individual data models and behaviors from the underlying specification components fit into an overall framework for a complete learning delivery environment.
Given the investment by the community in any version of SCORM, each subsequent version will require more in terms of support and additional elements that are not pure interoperability specifications.

SCORM evolution is also moving from more than just the inclusion of new standards and features into SCORM. Support documents, test suites, prototype implementations, etc., are all key components to be provided with each new release of SCORM and must be planned and coordinated with the technical evolution of the reference model itself.

The diagram also illustrates the formalization of specifications into de jure standards. As SCORM moves forward, each of the component specifications should be replaced by the corresponding formal standard. Such replacement, however, should not be used as a back door to add features to SCORM, but should focus on moving the SCORM bookshelf to formal accredited standards.

The diagram does not illustrate either the underlying model of the SCORM architecture, or how the SCORM architecture evolves as new component standards are added. The technical evolution of SCORM must be grounded in an architectural model of components and functionality, each supported by one or more specifications within the overall model. Such an architecture model provides part of the framework for moving forward.

**Generalized Learning Management System Architecture**

ADL has used the following diagram as an illustration of the model of a Generalized Learning Management System. The diagram illustrates a few of the key functions within a learning technology system including, those that are currently a part of SCORM.
SCORM 1.3 focuses on content management, tracking, and sequencing as part of the delivery of content. It defines metadata for use in describing content objects and content collections, and a format for the exchange of content between repositories and other content collections and learning management systems. The format includes descriptions of how to sequence and select content based on learner tracking and progress. SCORM 1.3 includes a tracking and control data model used at run time, and an API used for content to obtain control data from the learning delivery environment and pass back tracking results.

While the diagram illustrates some parts of a learning technology system, and the relationship of some of the parts to elements of SCORM, it neither shows all pieces of a learning technology system, nor does it show how these pieces can or should be related to underlying standards. Furthermore, it provides no guidance on how to move forward with new components.

It is important to recognize that the diagram explicitly assumes a service-based approach to the SCORM architecture.
Model of SCORM Evolution

The general model of components and content reuse and interoperability in SCORM has evolved through several generations of system designs. The series of diagrams below illustrate this process.

The same evolutionary approach, and versions of system design, has been used for other types of learning technology systems (e.g., intelligent tutoring systems) and in other disciplines. While each individual system, or types of systems, may not go through all of the steps, the sequence of diagrams illustrates the common stages of system designs and system evolution.

Learning System Evolution

Stage 1 -- Monolithic: Initial learning systems are designed as monolithic; the delivery system (providing content delivery and learning management) and the actual learning content are intertwined in a single custom product, intermixing content and control. Each course or learning application is designed and created separately as a unique system. There is no concept of separation of components or independence of content, no concept of content reuse and no application of learning technology standards. The diagram illustrates a collection of four different custom learning applications, each delivering a unique set of embedded content.

Stage 2 -- Separation of Content from System: Developers of a single system may recognize that their system design and delivery approach may be used in different learning applications. A second-stage system design begins to separate the learning content from the delivery system. In the first phases of separation, there is no unified model for content; only the idea that there are some common delivery attributes and features of delivery and management, and there are some other common attributes and features of content. An initial set of components may be created, operating on structured content.

Thus the content model (illustrated by the shape) and learning content (illustrated by color) is slightly different for each reuse of the basic delivery system. Reuse is beginning at the system level for each product and content offering, but there is not a clear separation of content from system, and each new learning application requires significant custom implementation. As illustrated, different systems (two of the systems from the diagram above) will begin to evolve in the same way, but each will take a approach to how the content and system are separated. At this stage, each delivery system has multiple implementations with slightly different content models.

Stage 3 -- Content Models: In the next stage, developers recognize that the same system model and content model can be applied across a range of learning applications. Systems evolve to have a common architecture or design, and to use a single content model for all learning applications. The same delivery system can used with new courses or content with no or minimal changes to the
system. While this type of design lets developers leverage their investment by producing only a single delivery system, their systems still rely on a proprietary content model. As the diagram illustrates, each different system evolves to use a different content model and representation. The benefit of the approach is primarily to the system and content developer; they leverage a single tool and approach. Common approaches to cross-system interoperability are not paramount.

**Stage 4 -- Common Content Models:** The fourth stage shows the first transformation from individual systems to a common approach: each of the different delivery systems adopts a common model for the representation of learning content; and has a set of common core delivery behaviors (content being processed and delivered in a consistent way across the different systems). Thus, while the design and features of each delivery environment may be different, allowing product differentiation, the systems operate using a common content model or representation and implement common, consistent behaviors. Learning content that matches the content representation could be moved between different delivery systems.

At each stage in the system evolution, the design, structure and components of each delivery system changes to match the requirements imposed by the changing content model.

In the stage illustrated above, there may be limited interoperability between systems. Different systems may adopt the same content model, but the agreement to use the same model may be made by a set of selected trading partners. The models themselves may be proprietary or closed and available to only a select set of partners. The diagram still illustrates the content as being part of the overall delivery system.

**Stage 5 -- Open Interoperability Standards:** The next stage, and most significant for interoperability, is to create a formal, neutral interface between the system and the content. Content is defined in a neutral, open format that captures aspects of content structure and behavior. The interface (represented by the dark green bars in the diagrams) and interchange format may provide models for content interchange and content interoperability. The same content model is used with all systems (illustrated by the common, separate shape), and different content can be used with different delivery systems (color illustrates a particular set of content).
The evolution of SCORM itself begins at the stage illustrated above. There is a content model (an aggregation or collection of the different parts of the model defined in the SCORM bookshelf) and various vendor and product offerings that use the content model. SCORM details the different parts of the content model and the design of the open public interfaces (using standards) between the content and the delivery and management environment.

**SCORM System Evolution**

The following diagrams expand the model of separation of content from delivery system. They illustrate how the content model is divided into pieces, using different interface and interoperability specifications as part of the overall design. They also illustrate a collection of four different interfaces, with corresponding models and interoperability standards (counter-clockwise from the bottom of the diagram):

- **content and content management** -- definitions of content and how to interchange content with delivery systems.
- **user interface and presentation** -- how to describe user interface features, navigational controls and content presentation.
- **learner management and enterprise integration** -- linking of the delivery system with enterprise functions and data exchange between learning systems and enterprise systems.
- **content delivery and control** -- managing actual content delivery and exchange of data between content and systems during learning.

Each of the subsequent diagrams expands on this first generic model, illustrated below, that shows how SCORM adds specific features, standards and capabilities to the model.

The diagrams and discussion below focus on only the generic features of the architecture and its components. They do not detail all of the elements needed to fully describe a system design or its implementation.
**SCORM 1.0:** The initial version of SCORM (SCORM 1.0) included limited models and features. It has a simple content and metadata model (only part of which is defined with one open interoperability standard). Content features and delivery are defined through the CMI model, encompassing both data and system communications. The delivery system uses its own private encoding of these data models. Limited content behavior is specified by the CMI model, and linked to content via the run-time environment (RTE).

The delivery system is shown to fully enclose the CMI content and RTE elements, along with its internal representation of the other models. SCORM specifies only the interfaces to the delivery system, and the behavior, not how the system is implemented. Thus each
implementation of a delivery system will choose its own internal models for representation and processing.

The diagram does not illustrate any other features or functions, e.g., assessments, content branching, navigation, presentation. While these may be present in a SCORM-conformant system, SCORM 1.0 made no statements about them and did not define their characteristics, behavioral requirements or interfaces. There may not be interoperability for or consistent behavior of the features not formally defined as part of SCORM.
SCORM 1.2: From an architectural view, SCORM 1.2 is a relatively minor evolution of the architectural design; the content model is formalized using the Content Packaging specification. This changes one interoperability interface.

While the architecture remains the same, improvements to the actual standards used, and clarifications of issues and resolution of problems significantly improves interoperability of those features included in SCORM 1.2. Also note that the changing models do impact actual implementations needed to conform to the evolving standards, even though the architecture remains the same. *Architectures, standards and implementations all may evolve independently.*
**SCORM 1.3:** The addition of sequencing in SCORM 1.3 extends the content model and interface for content exchange (adding sequencing rules and descriptions). A content delivery system must implement a prescribed set of behaviors defined in the sequencing specification (the specification defines how the sequencing rules and descriptions are used and processed, not how to implement or encode the processing). Logically, this adds the *Simple Sequencing Engine* (SSE) as an internal component of a delivery system. This is a logical addition; an explicitly encoded different system component is neither specified or required. The sequencing engine relies on implementation-specific encodings of the logical tracking and state models included in the specification.

**SCORM 1.x:** The approach to extending SCORM as illustrated above is generic. Additional features require:
- Extensions to the content models (defined by models from standards).
- Extension to the information used in by content during content delivery. Again, the data model for the information processed by the content is defined in the standards.
- Additions to the data models encoded in the delivery system.
- Encodings of the required additional behaviors defined by the added specifications.

Note, as each of the new features is added, the corresponding proprietary (and non-interoperable) solutions used in systems that conform to prior versions of SCORM are replaced by those defined by the added specifications. Thus systems and content become more interoperable as they evolve.

The following diagrams illustrate the addition of specific types of features to SCORM 1.x. Each of some potential additions are presented separately, and the order of presentation is not meaningful.

**SCORM 1.x with Assessments:** Descriptions of assessment items are added to the content models. A specification such as IMS Question and Test Interoperability (QTI) is used to exchange assessment descriptions between systems. Content objects will require assessment models to access data and record results and tracking, communicating over an extended delivery interface (possibly including specific functionality to support assessments).

A logical extension of the delivery system is the addition of an *assessment engine* (AE) (with associated tracking), and internal representations of assessment items. The assessment engine processes assessment data and tracks assessment results, linking these to other delivery behaviors.
SCORM 1.x with Simulation Content: Simulation content extensions are based on the addition of a simulation content model (used to describe the content and its exchange with other systems), a data model for content used at run time to control the simulation content, and a logical Simulation Engine (SE) within the overall delivery environment used to maintain simulation state and control simlets.
**SCORM 1.x with Competency-Based Objectives:** Formal models of competencies (competency definitions) can be used in a content model to define competency-based sequencing. Additional defined competency processing behavior (a *Competency Processor Engine* [CpE]) is required in the delivery system. Competency data is exchanged with enterprise and HR systems.
SCORM 1.x with Learner Profile-based Customization: Profile-based customization uses profiling descriptions defined within content, accessed via the content-to-systems communication interface, to customize the delivered content. Learner profile data comes from enterprise systems, via standard interchange models (e.g., LIP, AccLIP). Content models will include additional definitions on how to localize and internationalize content, and accessibility information. The delivery system must be able to process all of this information and will require internal data models. Customization processing is defined by a new behavioral component, a Customization Engine (CE).
SCORM 1.x with Navigation and Presentation Models: User interfaces, navigation, presentation, etc., are separate from content delivery and management. Additional specifications (to be defined) are needed, along with internal models used by the delivery system.
SCORM 1.x Redux: The following diagram is a combination of all of those above, illustrating part of the range of features and additions that could be added to SCORM 1.x. Some of the items in the delivery system have been renamed, e.g., the individual tracking and state models are replaced by more generic behavior components that maintain state and tracking results for different types of content.
Not all potential additions are illustrated, but many follow the same model. For example, formally supporting collaborative content would add a collaboration content model, a delivery and communications model and extensions to the delivery environment, including possibly a logical collaboration engine component.

The architectural evolution of SCORM 1.x follows the pattern described above. Additional individual features and components of the delivery system are split out from the existing proprietary solutions; content and other external models are split out or existing models are augmented; and the data models and behaviors of the new components are defined in the new standards added to the
SCORM Behavioral System Evolution

**SCORM Services Architecture:** The models illustrated above do not make any requirements on how to implement a SCORM-conformant system, only the behaviors that it must exhibit to the learner, and the data that it must be able to process. The key to interoperability is through data exchange and data interoperability standards, as illustrated in the simplified diagram below.
A true component-based architecture, with plug-and-play components, requires definitions of the behaviors and interfaces to each of the different components. The diagram below presents a behavioral alternative. Integration and interoperability are defined by the behaviors that components must exhibit, and defined behavioral interfaces between components.
In the diagram below, the general approach is expanded to reflect the overall SCORM architectural model from above. The delivery system is decomposed into a collection of individual components, each with a defined service behavior and interface. The actual SCORM delivery environment is then a control system that coordinates the other core SCORM services.
The diagram illustrates the final stage of a SCORM service model. The evolution to services could proceed piecewise: as a feature is added to SCORM, it could be added as just a set of behavioral specifications for the learning, data models and exchange standards (letting the delivery system decide how the feature will be implemented), or the addition could include a definition of a set of behavioral components with their defined interfaces. The latter still does not dictate how to build the behavioral components, only their granularity and interfaces.

Any of the potential extensions of SCORM 1.x could be illustrated using the approach presented above. The general model of
architectural evolution is to add new external content, model and data definitions, and define additional standard behavioral requirements on a SCORM-conformant delivery system.

Part II -- SCORM Issues

SCORM 1.x Issues

Issues to be considered in the evolution of SCORM 1.x fall into three major categories:

- Technical Extensions to SCORM.

During the development of the current SCORM 1.x series, users and developers have made requests for various technical extensions to SCORM, clarifications, modifications and additions to the suite of standards on the SCORM bookshelf. Key among these are:

- **Presentation** -- Models of separation of content from presentation.
- **Navigation** -- Models of separation of content from navigation controls.
- **Assessments** -- Support for assessment content.
- **Simulation Content** -- Support for simulation content.
- **Repositories** -- Models of content repositories and integration or interoperability of learning delivery with repositories.
- **Data Models** -- Enhanced tracking and data models.
- **Learner Profiles** -- Formal modeling of learner profiles.
- **Security** -- Approaches to insure data is not compromised.
- **Communication Models** -- XML/SOAP-based APIs and alternatives to the CMI API.
- **Web Services** -- Models using Web Services for interfaces and interoperability.
- **Collaboration** -- Support for collaborative and multi-user content.
- **Service-Based Content** -- Support for interactive learning services and service-based content (e.g., collaboration spaces, chat).
- **Localization and Internationalization** -- Supporting localized components and content, and international users.
- **Accessibility** -- Support for learners with different abilities and needs.
- **Content Variants** -- Models of version- and variant-specific content.
- **Digital Rights Management** -- Approaches to include DRM in SCORM.

- Learning Extensions to SCORM.

While SCORM was developed to support single-learner, procedural learning, SCORM could be extended to better support other types of learning and to better integrate with other systems. Major areas for potential extension include:

- **Competency-Based Learning**
- **Performance Support**
- **Mobile Devices and Mobile Learning**
- **Adaptive and Model-Based Systems**
Beyond the technical evolution of SCORM 1.x, and the addition of features and standards, other work needs to be done to provide the necessary infrastructure and support environment to evolve, use and deploy SCORM.

- **Infrastructure Extensions to SCORM and SCORM Bookshelf Management.**

Each of these are detailed below. There is overlap in the descriptions of the items as each is described fully and independently of each other, even though the issues are often interrelated.

Note: the order of presentation of issues is not an indication of the importance of the requirement, or the priority of developing solutions or addressing the issues in future versions of SCORM 1.x.

Each item begins with a general discussion of the item and includes a summary of key points useful in planning SCORM evolution. Note, the values presented in the actual issue summaries are best guesses. Detailed planning and background investigations that are needed before the issue is formally addressed may revise these estimates.

### Issue Summary

<table>
<thead>
<tr>
<th><strong>Complexity</strong></th>
<th>How complex is it to address the issue. The value ranges from simple or easy to difficult or complex.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Effort</strong></td>
<td>What is the expected level of effort needed to address the issue. The value ranges from small (weeks) to large (years).</td>
</tr>
<tr>
<td><strong>Impact</strong></td>
<td>What is the expected impact of successfully addressing the problem. Impact is for the learner or content developer, not the system developer. The value ranges from little or small to large or beneficial.</td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td>What is the expected scope of the solution. Scope is for both the systems and for content. The value ranges from isolated (small scope) to widespread (large scope).</td>
</tr>
<tr>
<td><strong>Research</strong></td>
<td>Is there an adequate research base to support a solution. Values range from none or limited through adequate or comprehensive.</td>
</tr>
<tr>
<td><strong>Practice</strong></td>
<td>Is there adequate demonstrated commercial practice to support a solution. Values range from none or limited through adequate or comprehensive.</td>
</tr>
<tr>
<td><strong>Standards</strong></td>
<td>Are there available specifications or standards that are in use or development to support a solution.</td>
</tr>
<tr>
<td><strong>Related Issues</strong></td>
<td>List of related issues that need to be considered in developing solutions to the issue. Issues are not referenced back to other issues where the issue is a supporting solution.</td>
</tr>
</tbody>
</table>
Technical Extensions to SCORM

Presentation

SCORM is currently silent on how to present or render learning content or the user interface (visual look and feel, interface skins). Each content developer and system implementor is free to choose their own styles and interfaces. Learners will find different interfaces and different styles of interaction if they move to a different system. Within any system or course reusing or aggregating content, learning content produced by different developers will look different and will use different forms and styles of interaction.

The problem for the learner is most severe when content from different developers using different presentation styles is mixed in reuse, e.g., mixed colors, layout, fonts, notations (e.g., rendering mathematics, references). Inconsistencies and differences in presentation and the ransom note effect will confuse the learner and reduce the effectiveness of the content.

Related to the actual rendering presentation is the management of glue with content, e.g., how to handle traditional cross-references and contextual anchors, such as "see table above", "as discussed in the last module", or "in the next lesson". Contextualized content, important to guide learners, is in direct conflict with context independence needed for widespread reuse.

An ideal presentation solution will separate learning content from learning presentation. While approaches such as style sheets can be used for XHTML web content, similar models are not available for other types of content. Style sheet approaches are adequate for managing look and feel consistency, but do not provide a solution to the glue problem.

Issue Summary: Presentation

Complexity Complex. Presentation influences all aspects of how content is managed and rendered. Glue and context conflict with reuse, so a solution must permit context to be added to context-free content. Additionally, content developers will reject solutions that limit their ability to control the learner's interaction with the content; they wish to exercise significant control over the learner's experience.

Effort Major. Investigate existing solutions; develop prototypes; develop guidelines or standards; explore use with real content. While a solution that is separate from other behaviors is desirable, particularly in a presentation specification, presentation is intertwined throughout content and it may be necessary to have it intertwined throughout an implementation. It may be difficult to layer presentation onto other models and behaviors.

Impact Significant. The glue and ransom note effect limit the ability to produce good content and good learning. Providing context is critical to content developers, as is reusing and reaggregating content and content collections. Organizations require models to brand their interfaces and content.

Scope Major. Systems may need to be redesigned to incorporate the selected presentation model(s) and standards. Content will have to be designed or redesigned to incorporate the presentation model. A migration path is required.

Research Limited. The problem has been repeatedly documented. Comprehensive solutions have not been presented.

Practice Limited. Several groups have begun to explore solutions such as Doc Book as a model for describing some of the aspects of content. There is no general content model or markup model for learning content. General approaches like SGML or style sheets have applicability but have not been explored or implemented broadly for learning.

Standards None. No learning technology standards exist for presentation, look and feel, or glue. Supporting core technologies such
SCORM currently contains only limited models related to navigation, e.g., it describes the requirements and behavior of certain events used in processing a sequencing navigation request. How the actual interface is rendered; the use of, labeling, and placement of navigation controls (e.g., forward, exit, logout); and inclusion of scaffold content controls (e.g., help, glossary) is left to the content developer or system implementor. Different implementations may choose to embed some navigation controls in content, while others place it outside of content.

Different forms of navigation, redundant navigation (both context- and system-rendered controls), or lack of navigation controls (neither the content nor the system provides controls as both assume the other is responsible to provide them) all confuse the learner and reduce learning effectiveness. The lack of a common approach leaves the content designer without any guidance on how to model the content user interface and no solid assumptions to build upon.

When reusing content from different sources, learners may be presented with conflicting controls and mixed interfaces, each with a different look and feel.

Navigation issues may also be related to sequencing and control issues, as developers may want to embed navigational controls within an element of content. Developers may also want to embed contextualized controls in their content. Contextualized navigation controls are in conflict with the concept of a SCO being standalone and independently reusable.

An ideal navigation solution will separate learning content from navigational controls.

**Issue Summary: Navigation**

**Complexity** Complex. Navigation influences all aspects of how content is managed, rendered and sequenced. Contextualized controls (both presentation and learning context) conflict with reuse, so a solution must permit navigation to be added to content. Interface designers will reject solutions that limit their ability to control the user interface; they wish to exercise significant control over the look and feel. A solution must support best practices for usability.

**Effort** Major. Investigate existing solutions, explore and document navigation controls and interface models used in vendor products; develop prototypes; develop guidelines or standards; explore use with real content. While a solution that is separate from other behaviors is desirable, particularly in a navigation specification, navigation is intertwined throughout content, presentation and sequencing behavior, and it may be necessary to have it intertwined throughout an implementation. It may be difficult to layer navigation onto other models and behaviors.

**Impact** Moderately significant. Some of the look and feel issues can be addressed as part of a presentation solution. Organizations require models to brand their navigational controls and interfaces. The most significant impact comes from modeling how navigation is embedded and contextualized within content. Providing context is critical to content...
Assessments

SCORM does not provide a formal model for design and delivery of assessments, or for interoperability of assessment content. Various approaches used include making each question or item its own element of content (which requires multiple learner interactions and substantial sequencing control) or making an entire assessment a single content element (which limits reuse of the pieces and eliminates detailed tracking and results data). Sequencing adds essential capabilities needed to design feedback and remediation patterns based on assessment results, but complex patterns require more data than the simple aggregate results used in simple sequencing.

Many groups have adopted IMS QTI as a basis for assessment. The QTI specification provides a model to describe assessment items and collections, and a detailed tracking and results model. It also provides some behavioral descriptions used to build assessments from their parts, e.g., selection and sequencing. QTI was designed primarily for information exchange and use in high-stakes testing. The SCORM learning model assumes that assessments are embedded and integrated with other content, and used to control content selection and learner paths.

Some implementations treat assessments as a separate class of content, using different delivery and control mechanisms (including being provided by a third-party assessment service). SCORM currently does not provide a model to mix and coordinate different content delivery processes.
Thus, while many of the pieces needed to have a formal assessment model and assessment support exist, or are partially available in SCORM, there is no complete integrated solution ready to be adopted. While a complete solution is needed, inclusion of QTI as an assessment modeling and data interchange approach may benefit the SCORM community. Inclusion of a standalone QTI-based model in SCORM is appropriate only if the QTI solution will be carried forward as part of a comprehensive solution.

**Issue Summary: Assessments**

**Complexity** Moderately Complex. A SCORM assessment solution must address diverse issues, including: interoperability and exchange formats for items, rubrics, collections and results; models of assessment selection and sequencing and their integration with IMS sequencing; models of assessment results and tracking and their integration with the CMI tracking model and the sequencing tracking model; models of using assessment services within content delivery.

**Effort** Moderate to Major. Solutions to a set of issues (outlined above) need to be developed. Elements to build a solution from pieces of other models exist.

**Impact** Significant. Assessments are essential in procedural learning. Creating interoperable and reusable assessment content, and developing realistic learning designs including assessment content is currently limited in SCORM.

**Scope** Moderate. Assessment design and creation is a well-defined part of the overall instructional design process, and thus...
In SCORM, simulation content currently can be treated and processed (e.g., described, sequenced, tracked, delivered) like all other content. Simulators often have more complex requirements for data tracking, in particular, state management. Some simple simulators operate through simple external coordination of individual parts of the simulator (simlets). SCORM's CMI data model and restriction on SCO-to-SCO communications limit the design of simulation content and use of simlets.

Scoping a solution to provide support for simulation content is one of the most critical aspects in determining an appropriate solution. Simulation requirements vary, and different approaches and models for use of simulation will result in different solution approaches. A simulation solution should also be limited to supporting simulation needs; other specific requirements and solutions should be developed for other content types as needed, e.g., a state model should be designed specifically to support simulation.

Issue Summary: Simulation Content

**Complexity** Moderate. If restricted to simple simulation content, simulator tracking models and data communications, simulation has only moderately complex requirements. Providing a new model for simulation or simlet control (beyond sequencing), being able to monitor (in parallel with their execution) and preempt simulators, or provide complex coordination of simulation with other content is very complex in the current SCORM architecture.

**Effort** Small. Simulation data models, tracking and simlet control can be supported with simple solutions.

**Impact** Limited to Very Significant. Simulation content is a specific or specialized type of content, being used less than traditional descriptive content or assessments. There is significant danger and significant negative impact from the wrong solution. The proposed IMS Simulation State Model is inappropriate in that it provides arbitrary, uncoordinated content-to-content communications. Abuse of the model will result in reduced content interoperability (rather than the 80MB SCO, the result will be the 80MB content package that cannot be reused, restructured or decomposed).

**Scope** Isolated. If limited to simulation content delivered and managed like other content, simulation requires only isolated changes to tracking data models. Adding more complex simlet management would result in a broad range of changes to
Repositories

The overall SCORM architectural model presents content as being delivered from a local storage or repository system, having been imported via a content package from some external source, such as an authoring tool or an external content repository. While the model assumes a local content representation used by the delivery system, the actual content need not be stored locally.

With the exception of the content packaging requirement for a content locator, and requirements for certain metadata elements on content, SCORM is silent on how to reference content, how to access it for delivery from a repository (local or remote), or how to locate it in an external repository. In addition, cross-domain security features of browsers prohibit the delivery of content from one system and access to data model elements from another, effectively enforcing the existence of a local content repository.

Forcing content import for each delivery system severely limits reuse and sharing of content, and makes content versioning and updating more difficult (it being more difficult to maintain replicated content). The lack of common models for content search and content repository interoperability further limit content reuse. Schemes for unique content identifiers (GUIDs, DOIs) are also missing but are needed to locate, share, manage and deliver content.

Content repository issues are also related to overall content management issues, including: content versioning, content localization and internationalization, and rights management (but a repository solution does not require solutions to these other issues). Independent of these, SCORM would benefit from a better model for how to reuse and deliver content from external repositories.

Research

- Mixed. Simple simulators and simulation content exists and there are adequate models of simulation behavior.

Practice

- Mixed. There is extensive practice in creating and using simulation content. There is limited documented experience in deploying simulator and simlet content in SCORM.

Standards

- None. There are no learning technology standards for interoperable simulation content.

Related Issues

- Data Models: Simulation tracking models are alternative data models. All potential data model changes should be considered together.
SCORM uses the CMI data model to provide data for content and to capture result and tracking data used to control content delivery. The CMI data model includes a diverse collection of features, ranging from general attributes that represent learner characteristics to those used for messaging between an instructor and learner for a particular instance of content delivery. While SCORM also has formal models for representing sequencing and sequencing tracking results used in content delivery, content can only directly access data from the CMI model.

Because of its historic precedent, the CMI model is weak in representing many attributes, including tracking results, learner attributes, assessment data and content state. Various proposals attempt to augment the CMI data model, break it into separate models, or rationalize parts of the model.

By itself, the CMI data model will not support the future needs of content and will not support other extensions and evolution of SCORM's learning features. The CMI data model needs to evolve in conjunction with the addition of other data models, and content must be able to directly access elements from other data models.

Issue Summary: Data Models

**Complexity**  Moderate. Data model elements individually are not complex, but the full range of elements and all of their interactions with different processes and behavior increases overall complexity in evolving the data model(s).

**Effort**  Moderate to Large. Data model evolution must be coordinated with the addition and modifications to other data models, and thus requires coordination over several work tasks. Work is needed to define new data model specifications, add different data model application profiles to SCORM, and restructure content selection, sequencing and delivery and other behavioral models to use the new data models. A migration path to the new models is needed, along with a plan to deprecate the older CMI data model attributes over time.

**Impact**  Moderate. Improved data models provide better information that content may use, and provide better results for controlling content selection and delivery.

**Scope**  Moderate. The scope for changing any single data model might be isolated, but addressing all potential changes and alternatives increases the overall scope.

**Research**  Mixed. Various proposals for data model changes have been developed and tested, but different elements of the solution are at different levels of maturity.

**Practice**  Limited. There is existing practice in using some of the other data models. There is no established practice in using...
Learner Profiles

SCORM does not have a formal model of learner profiles (learner characteristics, learner records such as portfolios or transcripts). The CMI data model has a small collection of learner attributes that may be used by content to control content presentation and delivery within the scope of an individual learning object. There are no global personalization methods, and there is no agreed upon way to exchange the learner attributes and profiles with any other system.

The work on standardizing the CMI data model is improving the definition of and rationalizing some of the learner attributes currently in SCORM, but the model still may not be adequate. A comprehensive learner profile model could be used to provide content customization, localization, accessibility and the basis for competency-driven learning and personalization. A learner profile model could also be used to track overall learning results across multiple courses and could be used as the basis for exchange of learner information between SCORM-conformant systems.

Issue Summary: Learner Profiles

Complexity    Easy. Learner profiles are relatively isolated in SCORM and the current procedural learning model.
Effort         Moderate. Adequate concepts and profiling models exist to be applied directly. Work is needed to add a learner profile specification to SCORM, create the application profile for the community, and provide a migration path to the new model (and deprecate the older CMI attributes over time).
Impact         Little to moderate. Simply adding an improved learning profile model will not result in any changes to how current content is processed and delivered. New content that uses the new profiling attributes will be more beneficial to the learner. There may be more indirect impact through localization, internationalization and accessibility.
Scope          Isolated. The CMI model attributes are replaced with the new profile model. An interchange model (possibly using content packaging) is added.
Research       Adequate. Sufficient background work has been done to create useful and stable specifications that could be added to SCORM. Prototypes that use the models for controlling and customizing learning have been created.
Practice       Limited. There is existing practice in creating complex learner models using available specifications. There is little practice in using these models in SCORM and little practice in the exchange of learner models between different
Security

SCORM is silent on how, if and when to enforce security, particularly when transmitting data. There are no formal requirements for securing data and results when a content object is communicating with the SCORM environment. Likewise, there is no way to ensure that data passed through any of the other interchange mechanisms (e.g., via content packaging) has not been tampered with during transfer.

Secure, tamper-proof data communications between content objects and systems is essential for some learning applications. The SCORM content communications infrastructure needs to support a secure communications model. Additionally, interoperability requires that the model be consistent across all implementations. Secure communications may require solutions for sharing and passing of identification, authentication and authorization credentials.

Simple XML messaging and data exchange is not secure from tampering. Senders, receivers and intermediaries can change the elements of the XML message or document without the knowledge of any other party. Encryption and digital signatures provide mechanisms to ensure that content tampering is detected and enables multiple components to trust the exchanged data.

Issue Summary: Security

Complexity Moderate. Individual elements of a security solution are straightforward. Combining all elements and building a complete solution that is fully secure in all aspects is more complex due to the interactions between components and technologies.

Effort Moderate. Some elements of a security solution (e.g., signed content packages) can be easily put into place. A complete solution will require more effort.

Impact Significant. A security solution is essential for those learning domains that require secure and trusted communications and data transfer.

Scope Isolated. Much of the security solution can be addressed as part of low-level, common infrastructure. It should not be apparent in content design or creation, or in system component development.

Research Mixed. A collection of security infrastructure elements is well established. The proper combination of items, or when to use particular approaches, may not be so well defined.

Practice Mixed. Secure content-to-system communications is used in some products. No standard interoperable approach (beyond using wire-level security such as SSL) is in place. There are no common solutions for trust in XML data exchange.

Standards Mixed. There are no learning technology standards that explicitly address security. There is a set of core infrastructure systems.
Communication Models

There is significant confusion over how to implement the SCORM API. Characteristics of the API and its use in the web environment cause additional difficulties, including problems in implementing cross-domain communications. Because of the confusion and misunderstandings around the API, there have been calls for creating alternative communications models.

Some implementors are suggesting a return to the simple AICC HACP protocol. Others advocate SOAP and XML-based messaging alternatives. Each of these may provide adequate basic communications, but each may also introduce other problems. Security is a concern with both alternatives. Transactional performance is a concern with SOAP, it being better for large-grained communications. SOAP is also designed primarily for server-to-server communications, not server-to-browser client communications, and thus may require substantial overhead and processing in a browser-based content presentation environment.

SCORM requires a simple, secure, robust, lightweight content-to-systems communications model. All of the requirements and issues surrounding the communications model must be fully documented and an alternative solution needs to meet all requirements.

Issue Summary: Communication Models

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Moderate. Complexity comes from not just providing a communications alternative, but addressing all of the related issues, e.g., security and efficiency.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort</td>
<td>Small. The effort in developing a communications solution focuses on a well-defined problem.</td>
</tr>
<tr>
<td>Impact</td>
<td>Significant. A solution will eliminate much confusion by system implementors.</td>
</tr>
<tr>
<td>Scope</td>
<td>Isolated. The problem is strictly scoped to how to implement the content-to-communications API. Conversion to an alternative solution will have a widespread scope and will require a migration strategy.</td>
</tr>
<tr>
<td>Research</td>
<td>Limited. There is limited research on what are effective alternatives (addressing all issues of security, efficiency in implementation and transaction costs, etc).</td>
</tr>
<tr>
<td>Practice</td>
<td>Limited. There is limited practice in using alternative approaches that meet all requirements.</td>
</tr>
<tr>
<td>Standards</td>
<td>Mixed. There are no ready-made alternative solutions. All elements needed to create an alternative are in place.</td>
</tr>
<tr>
<td>Related Issues</td>
<td>Security: Any alternative model must provide a robust, secure communications protocol.</td>
</tr>
</tbody>
</table>

Web Services

The books of SCORM are primarily descriptions of data models. There are some descriptions of what content and the delivery environment must do to support the data models, i.e., the resulting behavior that the learner should see, but there are no stated
requirements for how to build a SCORM-conformant system. Even though the SCORM architectural model is organized around services, implementations do not need to be built on services.

Lack of a component or service model yields implementations that are monolithic. One cannot acquire or build a SCORM environment from component pieces. The approach becomes more problematic as additional features are added to SCORM. Service-oriented abstraction boundaries define connections with abstract entities, but service-oriented implementations are required to build interoperable component-based systems.

If SCORM is to move to a true component architecture, then component structures need to be defined with specified behavioral interfaces; the interfaces will provide the necessary interoperability of the components. This in turn requires a complete, abstract service-based model architecture, including transport, messaging, discovery, description, security, transactions, etc.

Web services provide an abstract model of service implementations (abstract in that they can be bound to different implementations, e.g., WSDL/SOAP, REST, J2EE). Web service models can be used to describe SCORM components.

### Issue Summary: Web Services

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Moderate. Individual component web service models vary in complexity depending on the behavior. Various architectural approaches to web services introduce general complexity into any solution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort</td>
<td>Significant. Varying levels of effort needed for any of the components is related to the complexity of the individual components. Overall effort is significant as it requires systems to move to an overall service- and component-based architecture.</td>
</tr>
<tr>
<td>Impact</td>
<td>Limited. Web services have no direct impact on content developers, learners or learning; they are important only to those building, acquiring and deploying systems.</td>
</tr>
<tr>
<td>Scope</td>
<td>Major. A service-based architecture is a different fundamental basis for building a SCORM management and delivery environment.</td>
</tr>
<tr>
<td>Research</td>
<td>Mixed. Service prototypes have been built and service-based models and architectures have been produced. Different service-based architectural approaches are still being evaluated. Architecture fundamentals are still under development.</td>
</tr>
<tr>
<td>Practice</td>
<td>Limited. A few learning technology system components are available as web services. There are no complete service-based systems.</td>
</tr>
<tr>
<td>Standards</td>
<td>Mixed. There is developing work on service-based learning technology standards, and general approaches to web services models and architectures. Core web service standards exists. There is work in developing application profiles of the core standards as the basis for complete sets of interoperable web services. All of the work is still in the initial stages and may evolve with developing practice.</td>
</tr>
<tr>
<td>Related Issues</td>
<td>None.</td>
</tr>
</tbody>
</table>

**Collaboration**

SCORM content is designed to be instantiated for a single user, with one or more learning attempts per instantiation. Other content may be designed to support interactions and coordination of a set of learners, i.e., once the content is instantiated, all of the users...
have both a per user data model and a common shared data model.

While how to handle all the details of collaborations and data sharing and a general collaboration model are complex, simple models to define how content and content data models are to be instantiated (per user, per instance, per cohort) might help in supporting some types of collaboration.

### Issue Summary: Collaboration

**Complexity**  Moderate. If restricted to a simple model of content launch instantiation, and simple data models, collaboration has simple to moderately complex requirements. Providing a complete new model for collaborative learning is a complex addition to SCORM.

**Effort**  Small. Collaboration instantiation models and simple data models can be supported with simple solutions.

**Impact**  Limited to Moderate. Collaboration is a specific or specialized type of content, being used less than traditional descriptive content and assessments. A simple solution will have moderate impact for this restricted use, and will provide only basic interoperability and support of collaborative launch events.

**Scope**  Isolated. If limited to simple instantiation models, collaboration requires only a few isolated changes to SCORM content descriptions, delivery and tracking.

**Research**  Limited. A few proposals have been investigated in prototype systems.

**Practice**  Limited. There is no established practice in supporting collaborative content in SCORM.

**Standards**  Limited. The IMS Learning Design specification includes some elements of how to describe and instantiate per user versus per cohort content. Specifications are not available for other elements of a solution.

**Related Issues**  

**Service-Based Content**

Most SCORM content is static, i.e., it is authored before delivery or completely defined for dynamic generation. In the second case, the dynamically delivered content typically represents some form of customization from a pre-established base.

Learning applications often include other types of content, such as collaborative spaces or communication tools. This content can be classified as service-based. The content or interaction is not predefined, but controlled by the learner, either as a service provided to the learner, or as a collection of content dynamically built and maintained by the service.

In the SCORM content model, content locators can be used to point to services, but there is no additional model of how to invoke or use the service. The traditional content launch, delivery and tracking model of SCORM may not be applicable for all types of service-based content.

In addition, there are no standard definitions for the behaviors and capabilities for the common types of service-based content. While extensions to the general SCORM content model would describe how to link the service-based content into a course or delivery environment, there needs to be a defined set of baseline capabilities for each particular service and a set of interoperable guidelines for each type of service.
Localization and Internationalization

SCORM has limited, but evolving features to support localization and internationalization. SCORM must support use and deployment in many markets around the world. It must be possible to both localize (fit to the needs and norms of a locale, community, culture or organization) and internationalize (fit to the language) both SCORM systems and content.

All data models and specifications must support full internationalization and localization, e.g., use of multi-lingual strings. Internationalization work, like that underway as part of the effort to standardize the AICC CMI model, is key to providing internationalization. Data models need to describe the localization and internationalization attributes of content. Delivery systems must deliver the appropriate localized or internationalized content and support the corresponding interactions.

While significant internationalization support can be provided through the use of multi-lingual strings and support for extended character sets (e.g., Unicode), a set of core supporting technologies and approaches for localization is missing. Localization could be supported, in part, by providing localized content variants.

While localization and internationalization are usually considered together, the issues surrounding each are unique, and thus are listed separately.

Issue Summary: Localization
<table>
<thead>
<tr>
<th>Complexity</th>
<th>Moderate. Provide support for localized content selection, delivery, description and packaging.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort</td>
<td>Moderate. Each element of SCORM can be addressed independently, but core issues such as content representation require solutions to be in place to support multiple parts of SCORM. Work must proceed by first addressing core issues.</td>
</tr>
<tr>
<td>Impact</td>
<td>Moderate. Provides a direct solution to support localized content and systems. Localization may not be a pressing or a documented need, relative to internationalization.</td>
</tr>
<tr>
<td>Scope</td>
<td>Mixed. Some changes can be performed in isolation. Others, such as relying on a content variation solution or use of content packaging to support localization, require broader solutions. No changes need be made to existing content -- it will not be localized. New content will be tagged to indicate localization attributes.</td>
</tr>
<tr>
<td>Research</td>
<td>Limited. Solutions to content variations and versions could be applied. A complete set of requirements for localization has not been documented.</td>
</tr>
<tr>
<td>Practice</td>
<td>Limited. There are no widespread established common practice of how to localize content and systems. Most work focuses only on internationalization.</td>
</tr>
<tr>
<td>Standards</td>
<td>None. There are no learning technology standards for localization, and a common standard probably is not needed. There are no core underlying technologies. Learning technology standards do not formally support localization.</td>
</tr>
<tr>
<td>Related Issues</td>
<td>Content Variants: Localized versions of content may be represented as a content version or content variant. The solution that provides support for content variants might be used to support localization, or a more comprehensive solution might be required.</td>
</tr>
</tbody>
</table>

**Issue Summary: Internationalization**

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Simple. Ensure all data models support international character sets and multi-lingual strings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort</td>
<td>Small. Each element of SCORM can be handled independently, and the effort to add the internationalization capabilities to each standard can be quickly identified and proposed solutions can be readily put forth. Content developers should not need to do anything directly or explicitly to support internationalized content; content will be expressed in one or more international languages.</td>
</tr>
<tr>
<td>Impact</td>
<td>Moderately Significant. Eliminates problems with using SCORM in international settings.</td>
</tr>
<tr>
<td>Scope</td>
<td>Isolated. Changes are incremental to individual pieces. There are no crosscutting issues. There are no needed changes to existing content, or a direct migration process can be used to indicate appropriate character sets and encodings. New content will be internationalized in the content development process by tools.</td>
</tr>
<tr>
<td>Research</td>
<td>Established. Solutions and methods are well established in other disciplines. No background or development work is needed.</td>
</tr>
<tr>
<td>Practice</td>
<td>Mixed. Some specifications and SCORM components support internationalization. There are existing implementations in some tools and systems. Full international versions of all tools and support for all data models has not been demonstrated.</td>
</tr>
<tr>
<td>Standards</td>
<td>Mixed. There is no need for a direct learning technology standard for internationalization. Core infrastructure standards needed to build a solution are well established. Some learning technology standards already support internationalization.</td>
</tr>
<tr>
<td>Related Issues</td>
<td>None.</td>
</tr>
</tbody>
</table>

**Accessibility**
SCORM is currently silent on accessibility requirements and features. SCORM must support full accessibility features; all content must be accessible to all learners, and all system interfaces and controls must support accessibility features. While accessibility may be mandated for US government systems and content (US Section 508 requirements), accessibility is a worldwide concern. Good models for accessibility can benefit all learners and can be applied to all content and systems.

Data models need to describe the accessibility attributes of content and the accessibility requirements of learners. Delivery systems must support the appropriate accessibility features and interactions.

Issue Summary: Accessibility

Complexity Moderate: Ensure all data models support accessibility features. Requirements to support accessibility may need to be determined.

Effort Moderate. Each component of SCORM can be handled independently. While some accessibility solutions are in place, others need to be developed. Content developers will need to be aware of accessibility requirements.

Impact Moderately Significant. Eliminates problems with SCORM meeting accessibility requirements.

Scope Mixed. Some changes (adding accessibility attributes to individual data models) can be performed in isolation. Others, such as how to provide accessibility features for user interfaces, controls, etc., require broader solutions that are consistent across multiple components. No changes need to be applied to existing content -- it will not be fully accessible. New content will be tagged to indicate accessibility attributes.

Research Established. Requirements and solutions to describe accessibility requirements are well established.

Practice Limited. Accessibility features have been implemented in learning content on an ad hoc basis. Relevant specifications and corresponding implementations targeted specifically at learning and learning technology are just emerging. Core requirements (e.g., US Section 508) and general guidelines (e.g., W3C Accessibility Guidelines) are in place.

Standards Mixed. The Accessibility for LIP specification is close to release and provides a potential model for describing learner characteristics and requirements for accessibility. Proposed new work on accessibility extensions for metadata will provide a model to tag content and other objects. There is no work to describe how to apply the accessibility requirements and features in learning beyond general accessibility guidelines and requirements.

Related Issues None.

Content Variants

The SCORM delivery environment, and a content package, each provide support only for a single version or variant of each content resource. In many instances, there may be multiple versions (updates and modifications) and multiple variations of each version of a content object (e.g., language-specific versions, localized versions, delivery device- or platform-specific versions).

SCORM does not have a model or process to deal with content variants and versions. One approach is to ignore the content versioning and variants and leave it to external management of content repositories, e.g., management is handled at the repository level. Another approach is to require different content packages or organizations that correspond to each version or variant combination: a system would deliver only the specified content collection. This solution might be acceptable in some cases, but may result in a combinatorial explosion of options.
Proposals have been put forward to extend content packaging to be able to represent a content object and its variants for content exchange. The problem, however, extends beyond packaging and exchange of content. A total solution must address how to specify and select the appropriate content version or variant during delivery.

While superficially isolated, the approach used for content variants may also address other issues, including accessibility and localization. A solution should not be considered in isolation.

**Issue Summary: Content Variants**

**Complexity** Easy to moderate. Extending content packaging and content repositories to represent different content versions and variants is a straightforward addition to the model of a content versioning or variant structure. Beyond exchange of content, the delivery environment must be extended to specify and describe how to select the appropriate version and variant.

**Effort** Little to moderate. Extending content packaging requires a small amount of incremental work. Extending other components requires developing and adding the solution to the impacted elements of SCORM.

**Impact** Moderate. A solution may simplify other issues, including localization, accessibility and device independence. Content developers need not be concerned with variants unless they need to produce content object variants. Developers will be impacted if a content versioning model is put in place. Developers will need to know how to specify the run-time selection of versions and variants.

**Scope** Moderate. Versions and variants will require changes in packaging, repositories, selection and delivery (including tracking of the version or variant delivered). While each individual change may be small, the cumulative effect is larger.

**Research** Moderate. Models for versioning and variants are established in other domains. The full scope of how to describe, track, deliver and manage content versions and variants is less well documented.

**Practice** Mixed. Many implementations have adopted solutions to content versioning and possibly to content variants. There are no agreed upon common models and approaches to interoperability of versioned learning content.

**Standards** Limited. A proposal has been put forth for extending content packaging. Solutions to other aspects of the problem are not available.

**Related Issues** None.

**Digital Rights Management**

With the exception of the simple required *rights* fields in metadata, SCORM makes no statements about digital rights, intellectual property, permissions, etc., for content use or reuse. SCORM does not specify how, or if, a system should process rights expressions or how to enforce digital rights.

SCORM views rights management and enforcement as a business process issue, not part of the learning management and delivery process. While some organizations may handle digital rights as a trading partner issue outside of learning, other groups believe that addressing intellectual property issues is essential in establishing a *learning object economy*.

SCORM's treatment of digital rights as out of scope for learning may be valid for the US Government community, but the approach...
may not suit the needs of the larger community. There may be significant pressure in the future for SCORM to be explicit about
digital rights, digital rights management and digital rights enforcement.

Issue Summary: Digital Rights Management

**Complexity** Moderate to Very Complex. As a technology issue, digital rights is moderately complex. As a political and process issue, digital rights is very complex. Concerns over privacy, security and trust may result in digital rights management being embedded in networks and operating systems, thus influencing all applications and content, including learning, without concern for the requirements or needs of the domain. Digital rights is further complicated by patent and intellectual property rights claims surrounding the process of digital rights management itself and the by various underlying legal issues, such as differences in copyright statues and international accords.

**Effort** Major. Developing or adopting a solution that addresses the different concerns and competing interests will require effort on both the technical and political fronts.

**Impact** Moderate to Significant. There are some overlaps in digital rights expressions and sequencing and pedagogical constraints on content access. For example, digital rights constraints may express limits of how often and when content can be accessed, and instructional designers and organizations may place similar limits on what, when, where and how content can be accessed for learning. Redundant enforcement, or conflicting rights rules and enforcement may complicate learning delivery and content design. The development of a comprehensive approach to managing intellectual property may enable a significant learning object economy and spur substantial developments of learning content. Likewise, lack of a solution to intellectual property issues could limit development and adoption.

**Scope** Moderate. It may not be possible to layer digital rights and rights management onto learning technology without requiring fundamental changes in underlying systems. Depending on the nature of the solution, digital rights may become pervasive in all components and data models. Content and content development should not have to deal with digital rights directly.

**Research** Limited. While there is core background for digital rights, specific requirements and approaches that are valid in learning have not been documented.

**Practice** Limited. While there are prototypes and claims of complete digital rights systems, these solutions have not been generally deployed and there is no practice in embedding digital rights management into learning technology standards.

**Standards** Mixed. There are specifications and emerging standards for digital rights or restriction expressions and other component technologies (e.g., xRML, ODRL, SAML). There are no learning technology standards for digital rights and how the other general standards apply to learning is an open problem.

**Related Issues** None.

---

**Learning Extensions to SCORM**

**Competency-Based Learning**

In SCORM, a course is delivered without any knowledge of what competencies (knowledge, skills, abilities, mastered objectives) a learner has already acquired or what material should be delivered to the learner for the learner to meet specific objectives beyond those defined in the course. The same set of learning materials is delivered to all learners, regardless of background or needs.
Competency-based learning provides customized delivery based on the learner's known skills and competencies and the competencies covered by each learning object. While SCORM does not have a full model for competency-based learning, objectives in sequencing can be used within a single course to control competency-based content selection and delivery. There is no approach in SCORM to share competency results more broadly, or to formally map competencies to learning objectives and job roles, or create complex competency concept maps.

A complete system to provide competency-based learning will require additional features and capabilities not present in SCORM. A complete solution may require an additional set of standards or a different profile of existing standards.

**Issue Summary: Competencies**

- **Complexity** Moderate. Some elements of a solution are in place. Other elements can be added to the current overall SCORM model.
- **Effort** Moderate. While some pieces are ready to be added to SCORM, a complete solution is not available.
- **Impact** Moderate. A complete solution will have impact that may be of interest to a broader community, and competency-based learning models can be layered on conventional procedural-based learning and existing content objects.
- **Scope** Moderate. Competency definitions and competency mapping components are isolated additions to SCORM. Competency-based control and selection modifies sequencing behavior. Additional global tracking (or profile data) may be required. While the individual pieces are isolated, the total solution encompasses several elements.
- **Research** Limited. Prototypes have been created to do competency-based learning with SCORM. There are few models of concept maps used to control learning.
- **Practice** Limited. Some commercial systems provide competency-based learning, and use competencies and competency maps to track learners against them. While there are no standardized approaches for competency-based learning systems, some core infrastructure, such as interoperable human resources models of jobs and competencies, exists. Some communities have large collections of job and competency descriptions.
- **Standards** Limited. The Reusable Competency Definition specification provides a simple model for defining competencies. Human resources and knowledge management specifications provide additional information. Learning technology standards do not exist for other parts of the solution.
- **Related Issues** None.

**Performance Support**

SCORM focuses on procedural learning, targeted at an individual learner, working independently, in a formal learning setting. While not directly designed to support other types of learning and applications, elements of SCORM, and a generalized model of systems and content interoperability and content reuse could be used in different learning situations. Performance support is one such alternative learning model that could be supported by SCORM, or by the general notion of content reuse and systems based on interoperability standards.

Performance support provides just-in-time access to materials that may be repurposed or reused learning content. A performance support system includes a mechanism to find and access content, possibly with the search and access aided by information about the current use context (contextualized delivery of content). Performance support may be embedded into other task-based computerized environments.
The content management, content interchange and content repository features of SCORM may have applicability in performance support, along with information about the learner and associated competencies. A complete system to provide interoperability beyond content exchange and management will require additional features and capabilities not present in SCORM. A complete solution may require an additional set of standards or a different profile of existing standards.

**Issue Summary: Performance Support**

**Complexity**  Moderate to Complex. Identifying how to repurpose, reuse and manage existing learning content for use in performance support is relatively straightforward. Developing a complex performance support application profile may be as complex as SCORM itself.

**Effort**  Long term. Developing a complete performance support solution requires multiple activities and developments, including incremental releases and updates to a new application profile.

**Impact**  Small to Moderate. An isolated or partial solution may have little impact on how SCORM is used in procedural learning and little benefit in providing a performance support solution. A complete performance support solution will have more impact.

**Scope**  Small to Large. A partial solution based on content reuse for performance support will have limited scope, incrementally adding features to SCORM and allowing content to be reused in the different setting. A complete solution will require new systems and design of performance support environments that operate using the new application profile.

**Research**  Limited. Relevant background, architectures and models for interoperability in performance support need to be identified or developed.

**Practice**  Limited. While performance support systems are in use, there is little practice or design of general performance support systems based on interoperability specifications and sharing content (reuse and repurposing content from conventional learning systems).

**Standards**  Mixed. There are no learning technology standards for performance support. Existing learning technology standards could form part of an application profile for performance support.

**Related Issues**  None

---

**Mobile Devices and Mobile Learning**

SCORM is silent on the types of interface devices used to deliver content to the learner, and the types of connectivity between any delivery device and any associated learning management system. Content and systems are often designed assuming a browser-based delivery platform (with a conventional display) and assume continued connectivity and communications between the browser or client device and the learning delivery system.

SCORM needs to support both connected and disconnected learners, and learning can be targeted for a variety of delivery platforms, from desktop to laptop, PDA, or other mobile device.

Connectivity and device characteristics should be abstracted from content design. Synchronization and data transfer in disconnected use are general problems in other domains; a solution for learning and SCORM should rely and build upon generalized approaches and solutions, including general standards for mobile devices and device synchronization.
SCORM needs to be clear about the requirements for connected and disconnected use, and the requirements or assumptions of device-independent delivery. Aspects of mobile devices and mobile learning impact SCORM content presentation and navigation, and the mechanisms for content-to-system communications.

**Issue Summary: Mobile Devices and Mobile Learning**

**Complexity**  Moderate to Complex. Presentation, navigation, rendering and interface issues are complex. Providing a solution for communications and disconnected use may be less complex.

**Effort**  Major. Investigate existing solutions; develop prototypes; develop guidelines or standards; explore use with real content.

**Impact**  Little to Moderate. A mobile device and mobile computing solution will have little direct impact to the current SCORM community. A complete solution will have more impact and will be of interest to a broader community.

**Scope**  Significant. Systems may need to be redesigned to fully support mobile learning (there may be alternative implementations for mobile and connected use). Content will have to be designed or redesigned to incorporate mobile device interface and display models and content communication models. A migration path is required.

**Research**  Adequate. There is a collection of prototypes for mobile computing and mobile device interaction that could be applied to learning.

**Practice**  Limited. There are several SCORM prototypes and systems that deliver content to mobile devices. These are not built on common presentation models (there is no way to ensure consistent presentation of the SCORM content across these systems), and they do not have a common approach to data synchronization.

**Standards**  Mixed. There are no learning technology standards for mobile devices and mobile learning. Existing learning technology standards could form part of an application profile for mobile learning. Some core infrastructure standards (mobile networking, synchronization) that could be used to build a solution exist.

**Related Issues**

- **Presentation**: Mobile devices require solutions that address how content and interfaces are described in a device- and format-independent fashion.
- **Navigation**: Mobile devices require solutions that address how controls and interactions are described in a device- and format-independent fashion.
- **Communication Models**: Alternative content-to-system communication models may provide solutions for disconnected use.

**Adaptive and Model-Based Systems**

SCORM focuses on procedural learning, targeted at an individual learner, working independently. While not directly designed to support other types of learning and applications, elements of SCORM, and a generalized model of systems and content interoperability and content reuse, could be used in different learning situations. Adaptive and model-based learning systems is one such alternative learning model that could be supported by the general notion of content reuse and systems based on interoperability standards.

SCORM's procedural-based learning model uses predefined (created by the instructional designer) instructional paths and sequencing alternatives combined with limited tracking data. There is no standard mechanism for a system to create any response or learning pattern that is not preprogrammed or preplanned by the instructional designer.
SCORM's data tracking is factual, describing the materials delivered to the learner. SCORM does not have any models of what a learner knows or should know; only what content the learner has experienced, and individual and aggregate results from interactions with the content. Learning patterns cannot be described in terms of a learner model.

SCORM has limited models of user characteristics, such as job roles, and limited models of how to associate a competency, skill or knowledge objective with content and learning so that the system will automatically select and deliver the right learning content on demand. SCORM does not provide well-defined capabilities for content customization based on roles, skills or learning objectives.

Each of the above illustrates one way in which SCORM is different from adaptive and model-based learning systems that can produce appropriate customized content based on a model of the situation (role, objectives, needed materials) and a model of the learner's knowledge. Model-based systems may also use other more detailed representations of what the learner knows and how their knowledge is evolving when compared with a target model of what they should know at any point in the learning experience.

A complete system to provide interoperability and content reuse in adaptive learning beyond simple customization and predefined adaptive paths will require additional features and capabilities not present in SCORM. A complete solution may require an additional set of standards or a different profile of existing standards.

**Issue Summary: Adaptive and Model-Based Systems**

**Complexity** Complex. Adaptive and model-based systems are different from procedural learning and require many features and capabilities not present in SCORM or existing learning technology standards.

**Effort** Long term. Developing a complete adaptive and model-based learning systems solution requires multiple activities and developments, including incremental releases and updates to a new application profile.

**Impact** Little to Moderate. An adaptive and model-based learning solution will have little direct impact to the current SCORM community; the problem and learning approach is somewhat orthogonal to procedural learning. A complete solution will have more impact and will be of interest to a broader community.

**Scope** Large. A complete solution will require new systems that implement all of the required features. New content, courses and learning experience designs are needed to exploit adaptive and model-based learning.

**Research** Substantial. Adaptive and model-based systems are based on decades of core research in how to build systems and their required functionality. However, there is little work in translating this research from one-of-a-kind prototypes and pilots into a general learning system architecture and data models that provide interoperability and content reuse in learning (some models may be available in other AI domains).

**Practice** Mixed. Some of the adaptive and model-based research systems and prototypes have been deployed in practice and there are some commercial offerings, but there is little practice in providing interoperability and content reuse between systems. Many deployed adaptive systems can be characterized as stage 3 or stage 4 (as defined above).

**Standards** Limited. There are no learning technology standards for adaptive and model-based systems. Existing learning technology standards (learner profile, competency definitions) may provide small pieces of a solution, but many needed elements do not exist.

**Related Issues** Intelligent Tutoring Systems: Adaptive and model-based systems share many similar requirements with intelligent tutoring systems. A solution for both issues would be beneficial and may be less complex than two individual or independent solutions.
Intelligent Tutoring Systems

SCORM focuses on procedural learning, targeted at an individual learner, working independently. While not directly designed to support other types of learning and applications, elements of SCORM, and a generalized model of systems and content interoperability and content reuse, could be used in different learning situations. Generative intelligent tutoring systems is one such alternative model that could be supported by the general notion of content reuse and systems based on interoperability standards.

Tutoring is proven to provide substantially increased learning and performance by the learner. Individual tutors are skilled and tutoring is labor-intensive and expensive, making it difficult to scale. The holy grail of e-learning is to be able to provide the equivalent of a skilled individual tutor (one that dynamically generates and constantly updates an appropriate learning experience to meet the needs of the learner) at substantially reduced cost.

Individual computer tutors for specific domains have been constructed and found to be effective. They currently require significant development work, both to build the tutoring content (materials, strategies, assessments used to replace the individual human tutor) and an appropriate system design to apply judgment and strategies to select, deliver and monitor learner progress.

Tutoring systems are largely handcrafted, both at the system level and for the development of content and tutoring strategies (e.g., strategies may be hardwired into the tutoring engine). There has been little effort in developing general approaches and models for interoperability and reuse of content and strategies between systems.

Intelligent tutoring systems will require additional features and capabilities not present in SCORM. A complete solution may require an additional set of standards or a different profile of existing standards.

Issue Summary: Intelligent Tutoring Systems

Complexity Complex. Intelligent tutoring systems are different from procedural learning and require many features and capabilities not present in SCORM or existing learning technology standards.

Effort Long term. Developing a complete intelligent tutoring systems solution requires multiple activities and developments, including incremental releases and updates to a new application profile.

Impact Little to Significant. An intelligent tutoring solution will have little direct impact to the current SCORM community; the problem and learning approach is somewhat orthogonal to procedural learning. A complete solution will have substantial impact as it provides the basis for true generative, anytime, anywhere learning targeted at the specific needs of the learner.

Scope Large. A complete solution will require new systems that implement all of the features. New content, courses, tutoring strategies and learning experience designs are needed to exploit intelligent tutoring and create intelligent tutors.

Research Substantial. Intelligent tutoring systems are based on decades of core research in how to build systems and their required functionality. However, there is little work in translating this research from one-of-a-kind prototypes and pilots into general architecture and data models that provide interoperability and content reuse in learning (some models may be available in other AI domains).

Practice Mixed. Some excellent intelligent tutoring-based prototypes have been transitioned and deployed as successful commercial offerings, but there is little practice in providing interoperability and content reuse between systems. Many deployed intelligent tutoring systems can be characterized as stage 3 or stage 4 (as defined above).

Standards Limited. There are no learning technology standards for intelligent tutoring systems. Existing learning technology standards may provide small pieces of a solution, but many needed elements do not exist.
**Related Issues**

**Adaptive and Model-Based Systems:** Intelligent tutoring systems share many similar requirements with adaptive and model-based systems. A solution for both issues would be beneficial and may be less complex than two individual or independent solutions.

---

**Simulation Systems**

SCORM focuses on procedural learning, targeted at an individual learner, working independently. While not directly designed to support other types of learning and applications, elements of SCORM, and a generalized model of systems and content interoperability and content reuse, could be used in different learning situations. Simulation systems are one such alternative learning model that could be supported by the general notion of content reuse and systems based on interoperability standards.

Simulation systems are typically multi-user or multi-agent, may operate in distributed, networked computing environments (mixing users with agents and simulation hardware), and may operate in real time. Simulation systems often have complex behavioral models of real world components (either in hardware or software) that are proxies for the simulated element from the real world.

SCORM lacks capabilities for multi-user learning, real-time computing, distributed coordination between systems and learners, and behavioral models of components and elements in the simulated world.

Alternative simulation architectures, such as HLA, are in use in other disciplines. It is unclear how HLA relates to SCORM, e.g., can SCORM provide a general architectural model that subsumes multi-agent distributed simulation, or can traditional procedural learning be one element of a complex simulation environment.

Simulation systems will require additional features and capabilities not present in SCORM. A complete solution may require an additional set of standards or a different profile of existing standards.

**Issue Summary: Simulation Systems**

**Complexity** Complex. Simulation systems are different from procedural learning and require many features and capabilities not present in SCORM or existing learning technology standards.

**Effort** Long term. Developing a complete simulation systems solution requires multiple activities and developments, including incremental releases and updates to a new application profile.

**Impact** Little to Moderate. A complete simulation systems solution will have little direct impact to the current SCORM community; the problem and learning approach is somewhat orthogonal to procedural learning. A complete solution will have more impact and will be of interest to a broader and different community of users and learners.

**Scope** Large. A complete solution will require new systems that implement all of the features. New content, physical, world and agent models, courses and learning experience designs are needed to deploy simulation systems.

**Research** Significant. Simulation systems can be based on existing research and prototypes. HLA illustrates one model of translating this research from one-of-a-kind prototypes and pilots into a general architecture and data models for interoperability. There is little work in developing links between SCORM and simulation systems.

**Practice** Extensive. Many simulation systems for learning have been developed and deployed. There is existing practice associated with systems and content based on HLA. There is no practice in linking HLA-type simulations and large...
Gaming Systems

SCORM focuses on procedural learning, targeted at an individual learner, working independently. While not directly designed to support other types of learning and applications, elements of SCORM, and a generalized model of systems and content interoperability and content reuse, could be used in different learning situations. Gaming systems is one such alternative learning model that could be supported by the general notion of content reuse and systems based on interoperability standards.

Gaming systems rely on high-fidelity user interfaces with extensive user interaction. Games react to user interactions to adapt the user experience. A similar strategy could be used to adapt learning, embedded in a highly interactive environment.

Gaming systems also share many characteristics with simulation systems. They may be targeted at the individual, or may operate in distributed, networked computing environments (mixing users with agents). Gaming systems often have complex models of real world components. These models may be less accurate than in simulation; simulation requires high fidelity in modeling behavior; gaming often relies on fidelity of interaction at the expense of accurate behavioral models.

SCORM lacks capabilities for multi-user learning, high-fidelity interfaces and complex interaction models, distributed coordination between systems and learners, and behavioral models of components and elements in the gaming world.

Most gaming systems rely on a core gaming engine, customized per game. The core engine reduces development costs, but engines are not fully standardized and do not provide models to support reusable or interchangeable content.

Gaming systems will require additional features and capabilities not present in SCORM. A complete solution may require an additional set of standards or a different profile of existing standards.

Issue Summary: Gaming Systems

Complexity Complex. Gaming systems are different from procedural learning and require many features and capabilities not present in SCORM or existing learning technology standards.

Effort Long term. Developing a complete gaming systems solution requires multiple activities and developments, including incremental releases and updates to a new application profile.

Impact Little to Moderate. A gaming systems solution will have little direct impact to the current SCORM community; the problem and learning approach is somewhat orthogonal to procedural learning. A complete solution will have more impact and will be of interest to a broader and different community of users and learners.

Scope Large. A complete solution will require new systems that implement all of the features. New content (including models...
Enterprise Integration

SCORM focuses on learning delivery and content interoperability. Learning delivery occurs within a managed environment, e.g., an LMS. While SCORM assumes managed learning, management focuses only on tracking of content that is delivered to make decisions about delivery and control the learning experience. SCORM is silent on management of learning and integration of learning systems with enterprise systems and back-office systems.

In the current SCORM environment, there is no interoperability between individual learning management systems outside of content interoperability. An organization deploying multiple learning systems will have unique methods to register learners in each deployed system, to pass summary results back to the enterprise HR systems, to obtain results and reports, etc. Each deployment requires unique and possibly custom integration into the organization's enterprise systems.

Learners encountering multiple systems see learning as disconnected experiences. There is no coordination of learning between systems and courses (e.g., tracking results for the same objective in multiple courses delivered by multiple platforms are not linked). There is no way to seamlessly embed learning delivered from different systems into an overall computing environment.

Moving beyond disconnected systems and disconnected learning experiences requires integration with enterprise systems, and integration and data sharing between learning systems. This requires additional interfaces, behavioral interoperability and data models beyond those for content interoperability and content delivery.

Issue Summary: Enterprise Integration

Complexity  Moderate. The types of data models and behaviors are individually simple. Complexity comes from the range of required models to provide a complete solution. Elements of the solutions may be provided incrementally.

Effort  Medium to long term. Some elements, based on existing solutions, can be added to SCORM directly. Other elements requiring new specifications and models will require additional effort over the long term.

Impact  Little to Moderate. Learners may not see a direct impact in their learning, but may find it easier to operate in an
Environment with multiple systems or embedded learning. The impact on management of learning and an enterprise deployment will be more important with reduced system deployment and integration costs, even for organizations that deploy only a single learning system.

**Scope** Isolated. The work is separate from the learning components of SCORM and in general can be treated as an orthogonal set of features.

**Research** Not applicable.

**Practice** Established. There is established practice, including use of interoperability specifications, in some areas such as moving registration lists and summary results between systems. Many other systems support enterprise integration, albeit possibly using an ad hoc approach. Generalized approaches and a complete range of solutions are not in use.

**Standards** Mixed. There are some enterprise specifications for learning (e.g., IMS Enterprise) and other enterprise specifications outside learning (e.g., HR specifications and data models) that could be applied.

**Related Issues**

**Web Services:** Behavioral integration could be based on web services models for component integration.

### Infrastructure Extensions to SCORM and SCORM Bookshelf Management

#### Semantic Web

SCORM is based on a set of data models and behaviors that are defined in the normative documents and elements of the SCORM bookshelf. Elements of SCORM by themselves have no meaning and are not self-descriptive, e.g., a numeric value that is a score only makes sense as a score by the context in which it is used. There is no fundamental set of semantics or model of the actual components that make up a system or content.

The semantic web is a general approach to developing models with rich meaning (each item is uniquely identifiable and has a name and a related set of named attributes, each of which is also an element in the web of items). This model permits the development of complex information and knowledge structures, and lets systems reason about the objects and their structures directly, not just relying on the implicitly encoded semantics in the content and corresponding programs.

The semantic web, and semantic modeling, offer a potential approach to creating highly flexible systems and self-reasoning systems. While useful in creating any complex, adaptive and intelligent system, the approach may be most valuable in creating intelligent tutoring systems and in developing a basis for a SCORM 2.x architecture.

The semantic web is one approach to semantic modeling. It currently relies on RDF data representations and associated processing and ontology systems such as DAML+OIL. The semantic web operates over the core web infrastructure. Efforts are underway to create a semantic web stack to define the layers of behavior in semantic processing.

In many ways, elements of the semantic web are similar to web services elements, and there are some analogies between the two service stacks. The semantic web model is still evolving, and efforts are underway to reconcile some of its characteristics with the general web and web services models, the latter currently being more broadly adopted in practice.
Due to ongoing developments in the field, the potential for applying a semantic web approach to SCORM to produce broadly interoperable, standards-based learning is not clear.

**Issue Summary: Semantic Web**

**Complexity**  
Very Complex. The semantic web provides a different modeling and architecture approach for building systems. Many elements have to be completely reconsidered. The full range of elements and their interrelations necessary to use the semantic web for learning in an interoperable way is not clear.

**Effort**  
Major / Very Long Term. Building a SCORM approach on the semantic web requires starting with fundamental research; developing and testing basic concepts and applying them to learning; moving from concepts to learning prototypes; and then finally to guidelines, specifications and standards.

**Impact**  
Significant. Semantic models enable other systems to reason about a given system and set of data, enabling the creation of complex adaptive and intelligent systems. A fundamental approach and architecture is needed to develop future generation systems.

**Scope**  
Large. Semantic approaches will completely restructure content and systems. There may be no simple or clear migration path.

**Research**  
Mixed. Semantic modeling concepts and research systems based on semantic modeling are part of the long history of artificial intelligence. Core research in the semantic web is still underway. Early research prototypes of using semantic web concepts for learning are just emerging.

**Practice**  
Limited. There is no established practice of migrating research and prototypes of systems using the semantic web into commercial offerings. There is no established practice in providing interoperability and content reuse between systems using semantic web concepts.

**Standards**  
Limited to Mixed. There are no learning technology standards for the semantic web. Some core infrastructure standards exist and may be useful (DAML+OIL, RDF), but this core infrastructure is still evolving.

**Related Issues**  
Web Services: The semantic web shares some core issues with web services; common approaches may emerge.

**Versioning and Version Management**

With each new release of SCORM, there will be content, metadata and packages that will conform to the older versions. New versions of SCORM may not be fully backward compatible with the previous versions. It is unreasonable to assume that all materials that conform to older versions will be migrated to newer versions of SCORM. Rather, there will be content packages that will mix content objects designed to conform to different versions of SCORM, and delivery systems and tools will be expected to process and deliver content that conforms to the multiple versions.

For components and systems to successfully process content that conforms to the different versions, the content must be clearly tagged to indicate version. Processing components need to be able to detect the version(s) of content and data. Those using SCORM systems and tools need to know what versions of content their tools and systems can process.

Beyond knowing the versions of the specifications on the SCORM bookshelf, differences in underlying versions of the supporting standards (e.g., versions of XML Schema) may also influence how content is represented and processed.
Systems need to be able to locate all versions of schemas and supporting control documents. These must be maintained in a stable environment and must be maintained and accessible over the long term.

SCORM requires a well-documented, stable set of versioned control documents and a documented strategy to specify and detect versions of content and components. The strategy must include how to specify versions for all elements and how the SCORM processors will deal with multiple versions of content.

Issue Summary: Versioning and Version Management

| Complexity  | Easy to Moderate. Existing versioning solutions may provide adequate solutions. |
| Effort      | Moderate. Specifying how to version schema and content and how to maintain control documents is straightforward. Developing strategies to deal with content that mixes conformance to multiple versions requires more effort. |
| Impact      | Significant. A successful versioning strategy will permit content that conforms to various versions of SCORM to be mixed and matched during delivery. Systems will be designed to support multiple versions of content in a consistent manner. |
| Scope       | Limited. Versioning is specific to tagging content and systems, and resolving how to mix and match different versions of materials. Versioning does not impact overall content design, design of learning strategies or general learning facilities and capabilities in SCORM. |
| Research    | Adequate. Sufficient background work has been done to create a solution for SCORM. |
| Practice    | Mixed. Versioning and version control are well established in practice for many types of information systems and data models, but there is no established versioning practice for interoperable learning content and systems. While there are general established methods, versioning and version interoperability remain problematic in many domains (e.g., Windows' DLL hell). |
| Standards   | Not applicable. |
| Related Issues | Migration and Backward Compatibility: Backward compatibility and version migration are versioning issues specifically related to maintaining and developing content and components. |

**Migration and Backward Compatibility**

Newer versions of SCORM may not be fully backward compatible with the previous versions. Changes to the data models, schemas (additions, depreciation, name changes, moving items between different name spaces) and API function names will cause interoperability problems when mixing versions of content that conform to different versions of SCORM.

Support for content migration, including transition tools and interface layers, will provide for backward compatibility between versions and methods to migrate and translate content between versions. Potential tools include data instance translators (mappings between schema versions, proxy services and API layers).

Issue Summary: Migration and Backward Compatibility

| Complexity  | Easy to Moderate. Mechanical translation (XSL translators) are relatively easy to produce. Translation tools that require semantic analysis of content are more complex. |
Several IMS specifications (IMS Simple Sequencing, Learning Design, Content Packaging and Question & Test Interoperability) have overlapping features and represent multiple, differing approaches that can be used to represent the same problem. In some cases, elements of these specifications are in conflict. In other cases, the differences are essential, but the relationships among the specifications and the rationale for the differences may not be obvious. There have been calls to harmonize the approaches across different specifications. The resulting changes to one or more of these specifications may impact how they are profiled and used in SCORM.

Harmonization that provides a consistent approach or eliminates redundant features can be beneficial to SCORM. Harmonization that eliminates features needed by SCORM will be detrimental.

Since the different specifications are built on different models (which are not fully documented), are subject to reinterpretation and suffer from a loss of historic justification and prior decisions, rational harmonization may be difficult.

**Issue Summary: IMS Specification Harmonization**

**Complexity** Easy to Moderate. Harmonization that is consistent with SCORM should be easy to apply. Harmonization that breaks features or requires substantial extensions to SCORM to replace the lost behavior could require complex modifications and workarounds.

**Effort** Simple. Most harmonizations that do not negatively impact SCORM should be simple and easy to implement.

**Impact** Little to Moderate. Harmonization that is consistent with SCORM should not have substantial impact. Harmonization that breaks features or requires substantial extensions to SCORM to replace the lost behavior could have significant impact.

**Scope** Simple. Most harmonization should be independent of other aspects and relatively isolated. Resulting changes should be specific and isolated.

**Research** Not applicable.

**Practice** Limited to Moderate. Some calls for harmonization are based on actual practice. Others are more theoretic and may lack supporting practical experience and implementations. There is a danger in calls for harmonization that are not
Emerging Specifications

Different communities are developing new learning technology specifications that may be of use in SCORM. In some instances, organizations are specifically suggesting that their work will be of interest to SCORM (or should be included in SCORM). Other emerging specifications, if targeted for SCORM or not, might be useful additions to SCORM overall or as part of the SCORM infrastructure.

Most of these efforts in producing new learning technology specifications are disconnected, and are not part of a comprehensive plan. While the individual pieces may be of value to SCORM, the overall impact cannot be judged by considering the pieces in isolation.

Potential candidates for the short term consideration include:

- Modification to Content Packaging to support content variants (discussed as an individual issue).
- Support for simulation content (discussed as an individual issue).
- Web services models (discussed as an individual issue).
- Repository integration and object identifier models (discussed as an individual issue).
- Vocabulary representation models (IMS VDEX). Vocabulary representations may be a useful element of common infrastructure for SCORM (e.g., equivalent to using XML or Schema models to represent data models).
- Interactive content (IMS Interactive/Next Generation Content). Use cases are all being put forward that may result in new content models that in turn may provide additional support for new types of simple content; new or modified data models, tracking or sequencing extensions; and more comprehensive models to support adaptive delivery.
- E-Portfolios. Plans are underway to develop an e-portfolio specification that could be used to record a comprehensive history of all learner interactions (lifelong, across multiple courses, multiple course providers and organizations, etc.). E-portfolios may provide more detailed information for content customization and personalization or may present new requirements for storing and maintaining results of learner interactions.
- Metadata profiles. Multiple metadata profiles exist. A common approach to creating a profile (including specifying extensions and customized vocabularies) will increase interoperability of content created under one of the many profiles.
- E-Reserves. In academic settings, a course is often accompanied with a set of readings; these are often placed on reserve in the library. E-Reserves provides an information model and exchange mechanisms for such lists. Use of the information on the list (there may be access restrictions) and its association with parts of a course parallel some aspects of auxiliary resources in sequencing.

It is critical that the addition of new components to the SCORM bookshelf be driven by user requirements. In considering any potential emerging specification, it is also critical to evaluate how to best meet the expressed needs. Solutions may vary from a full new specification, to coordinated extensions to one or more existing specifications, or a solution that is a profile or just an appropriate use of existing specifications. The selected approach must consider and balance technical issues, overall community impact, costs, efforts, etc.
Specifications into Standards

As an application profile, SCORM is based on a collection of formal standards and industry specifications, with ADL- and SCORM-specific extensions and profile constraints. As SCORM has evolved, the focus has been on using specifications and consensus material from external sources rather than ADL-developed components.

Certain elements of SCORM are emerging as formal standards and other components are entering the formal standardization process. Industry and adopters prefer that SCORM be based on formal standards, or based on components moving towards standards whenever possible.

Metadata is established as a formal standard. Elements of the CMI data model are also close to being established as formal standards. More elements of the CMI model need to be moved to formal standards, including a set of standardized behaviors associated with the data model, a binding of the data model items to XML, and a binding of the data model to the API.

Content packaging is ready to enter the standardization process, possibly as two elements: a data model and a binding. However, some aspects of content packaging (variants, submanifests) may not be ready for standardization. Additionally, there are potentially competing packaging models (content packaging, MPEG-21, METS). Overall general computing industry forces (market size relative to learning, adoption in other market segments) may make switching to one of the other content packaging alternatives an appropriate decision.
Other components of SCORM, such as sequencing, are not well enough established in practice to enter standardization at this time. Specifications that could be added to SCORM should also undergo a period of use and evaluation as part of SCORM before being standardized; industry use in SCORM provides valuable feedback that is useful in standardizing existing specifications.

One danger in the standardization process is using the process to rewrite and modify a specification, rather than codify and formalize it. Another danger is in overloading individual standards with multiple behaviors. Splitting specifications into multiple parts, each addressing a clearly defined component may require more effort in standardization, but individual components may be more easily used, modified and profiled.

Issue Summary: Specifications into Standards

**Complexity**  
Easy to Medium. Standardizing an individual component is not technically complex. Complexity arises from maintaining consistency in approach and harmonizing approaches across multiple standards.

**Effort**  
Ongoing. Effort on standardizing any individual specification may be short to medium term once the work commences. Some work is underway. New work will continue as more components move to standards and more components are added to SCORM.

**Impact**  
Moderate. Moving from specifications to standards has little effect for actual learners and content developers; they only care that there is interoperability, not how it is achieved. There is more impact in overall industry adoption and perceptions about the credibility and reliability of SCORM. Substantial modifications to a specification as part of the standardization process may impact existing systems and content. A migration path is required.

**Scope**  
Isolated. Most components of SCORM can be standardized in isolation from each other.

**Research**  
Not applicable.

**Practice**  
Established. Current standardization efforts have established and demonstrated the procedures and processes that can be used in further standardization efforts.

**Standards**  
Not applicable.

**Related Issues**  
Emerging Specifications: Elements to standardize are influenced by new specifications and changes to existing specifications.

---

**Part III -- SCORM Futures**

**The SCORM Bookshelf**

SCORM explicitly includes several standards in the application profile. However, as a whole, SCORM relies on a much richer set of standards and other key elements. Explicitly recognizing all of the components, tracking their evolution, and formalizing the relationships between the components is beneficial to the evolution of SCORM. One approach to present all of these elements is to elaborate the SCORM bookshelf model.

The overall model can be divided into four collections of the books of SCORM.

- The *Standards Bookshelf* is the collection of the learning technology standards that make up SCORM.
- The *Environment Bookshelf* is the collection of pieces of SCORM used to create the overall technology structure around
SCORM-conformant learning systems.

- The **Content and Application Bookshelf** is the collection of information about how to apply and use SCORM in learning, focusing on content development, use and SCORM deployment.
- The **Core Infrastructure Bookshelf** is the collection of general technology standards that underlie all of the other standards in SCORM and its overall technology infrastructure.

The elements of each of these bookshelves are outlined below, followed by a presentation of a general model.

**SCORM Standards Bookshelf**

The SCORM 1.3 bookshelf contains the following elements:

- The **Metadata Book**. The metadata book is built from two elements.
  - Learning Object Metadata (LOM). IEEE LOM 1484-12.1-2002 is profiled in SCORM to provide metadata for all items including content objects and content packages. LOM provides a description of the items within a metadata record instance, without specifying a processable or interchangeable representation of the data.
  - LOM XML Schema Language Binding. The IEEE LOM XML schema language binding is used to provide a machine processable representation of a metadata instance.
- The **Content Communications API Book**. The content communications API book is built from one element.
  - IEEE API. IEEE CMI API 1484.11.2-2003 is profiled in SCORM to provide the mechanism for a SCO to communicate with the learning management system, to get data used by the SCO, and to report results back for tracking.
- The **Content Data Model Book**. The content data model book is built from two elements.
  - IEEE CMI Data Model. IEEE CMI Data Model 1484.11.1-2003 is profiled in SCORM to describe the data used by a SCO. The IEEE data model provides only a description of the data. The data model must be combined with a set of behavioral descriptions of how the data is used within a learning management system and how the data is communicated over the API.
  - AICC CMI Data Model. The AICC CMI data model includes the details of how the IEEE data model is used by a learning management system and how the data is communicated over the API. (As part of the evolution of SCORM from being based on specifications to being based on formal standards, more formal standards will replace the AICC specifications.)
- The **Sequencing Book**. The sequencing book is built from two elements.
  - IMS Simple Sequencing Information and Behavior Model. IMS Simple Sequencing (SS) 1.0 is profiled in SCORM to provide the description of a set of rule-based behaviors used to sequence the content in a learning experience. The SS Information and Behavior model includes both a description of the information used to describe a learning sequence and the behaviors that a system must exhibit when processing the sequencing instructions. By itself, the SS information model does not prescribe an interchange model or external representation.
  - IMS Simple Sequencing XML Binding. The IMS SS XML Binding is used to specify how to include an instance of an SS 1.0 information model in an IMS Content Package.
- The **Content Packaging and Interchange Book**. The content packaging book is built from two elements.
  - IMS Content Packaging Information Model. IMS Content Packaging (CP) 1.1.3 is profiled in SCORM to provide the description of how to exchange data with learning technology systems. CP includes a representation of the structure of the content and its associated metadata.
  - IMS Content Packaging XML Binding. IMS CP XML Binding is used to specify the external representation of all content description data (i.e., non run-time tracking and control data). The basic IMS CP binding structure is augmented with the description of sequencing data represented using the Simple Sequencing XML Binding and profiled with the addition of SCORM-specific elements.
As described above, many of the solutions to many of the issues needed to evolve SCORM 1.x will result in new specifications and standards and additions to the Standards bookshelf. Potential candidate books include:

- The *Assessment* Book.
- The *Learner Profile* Book.
- The *Simulation Content* Book.
- The *Collaboration Content* Book.
- The *Service-Based Content* Book.
- The *Repository* Book.
- The *Navigation* Book.
- The *Presentation* Book.
- The *Security* Book.
- The *Localization and Internationalization* Book.
- The *Accessibility* Book.
- The *Content Management* Book.

These books might be developed individually, or the information may be grouped together into appropriate logical collections.

SCORM 2.x may require additional books for each of the added types of learning:

- The *Competency-Based Learning* Book.
- The *Performance Support* Book.
- The *Adaptive and Model-Based Learning* Book.
- The *Intelligent Tutoring Systems* Book.
- The *Simulation Systems* Book.
- The *Gaming Systems* Book.
- The *Mobile Devices and Mobile Learning* Book.
- The *Enterprise Integration* Book.

**SCORM Environment Bookshelf**

The *SCORM Environment Bookshelf* includes the documents needed to develop, test and deploy SCORM-conformant systems (not content). Elements of the bookshelf, which derive from the books on the Standards bookshelf, include:

- The *Test and Conformance* Books, including the actual test suite software, guidelines for its use, and the formal SCORM-conformance test criteria.
- The *Prototype* Books, including all of the sample and prototype implementations of SCORM systems, and the collection of demonstration and prototype content for developing and testing SCORM systems.
- The *Versioning and Migration* Books, including documents on SCORM's versioning strategy and tools used for migrating content between versions.

In addition to the books, the overall SCORM environment includes the control documents, and the ADL sample content repository.
SCORM Content and Application Bookshelf

The elements of the SCORM Content and Applications Bookshelf fall outside of the technical scope of SCORM, but are derived from the elements of the Standards bookshelf. They focus on how to develop, use and deploy SCORM for learning (at the content level, not how to build a SCORM-conformant system). Elements include design guides, authoring guides, content programmer guides, and information about SCORM content repositories.

SCORM Core Infrastructure Bookshelf

Underlying all of the elements of SCORM is a common set of Internet, Web and information technology standards, the SCORM Core Infrastructure Bookshelf. Key elements of the core bookshelf include:

- The Internet Core Specification Books, including core Internet protocols, selected RFCs, selected ISO standards, character sets, etc., used in learning technology standards.
- The Web and XML Specification Books, including core XML schema and data modeling technologies (e.g., XML, XML Schema, XPath, XML Query) used to define data and information models that make up various learning technology standards, and the XML languages used within content (e.g., SVG, SMIL, XHTML).
- The Content Specification Books, used to catalog and represent content in repositories.
- The Security Specification Books, used to maintain content and system security. Specification include the XML technologies used in security (e.g., XML Signature and XML Encryption) and specifications from OASIS (e.g., SAML, XACL) and related security technologies (e.g., PKI).
- The Digital Rights Specification Books, including DRM specifications such as xDRL and ODRL.
- The Web Services Specification Books, include web services technologies (e.g., SOAP, REST, WSDL, WSCL, UDDI) and application profiles used to create interoperable web services (WSI).
- The Accessibility Guidelines Books, including the W3C WAI Guidelines and US Section 508 criteria.
- The Learning Technology Architecture Books, defining how to build general learning technology systems, methods to profile learning technology specifications for a particular community, and web services models for interoperable learning technology web services.

The Bookshelf Stack

All of the books, and the corresponding bookshelves, can be arranged as a four-shelf bookcase, or stack of bookshelves, as illustrated below. The books and specifications on each of the bookshelves builds upon those from the lower shelves.
It would help in the understanding of SCORM if all of the relationships among the various books and underlying standards and specifications were explicit. System and tool developers in particular need to know about the underlying specifications and technologies to successfully implement SCORM-conformant systems. Content developers need to know about what technologies are supported by SCORM-conformant systems to build content that can be successfully deployed. Also, as the supporting elements at any level are updated and new versions are released, developers (both system and content) need to know how the changes might impact their work.
The illustration is not meant to be definitive; additional components will be added over time. Books on the Environment and Content and Application bookshelves need not all come directly from the ADL initiative, but should be developed in conjunction with the entire ADL community.

**Criteria for Inclusion of Components in the SCORM Bookshelf**

Decisions on when and if to add an additional book to the standards bookshelf of SCORM should be based on a rational set of criteria. Beyond having an established need and requirement, and a potential solution (based on research and prototypes), the following criteria could be applied.

- Is there an established consensus specification, i.e., a specification developed by an open group, rather than a proposed proprietary solution or a research model? The specification development group should have formal processes and have a model for group membership (not a closed, private group).
- If there is no specification, defer work until a specification has been developed by external groups.
- Are the specification and implementation that use it unencumbered by intellectual property constraints?
- Are there multiple independent implementations of systems that are based on the specification (e.g., 3 or more)? While no implementation needs to implement all features, actual implementation experience is essential. There should be multiple implementations of all core features. Independent implementations (each having a different code base) are used to identify differences in interpretation and differing assumptions. Multiple implementations also indicate that the solution may be widely adopted when added to SCORM.
- Are there multiple collections of content developed based on the specification (e.g., 3 or more)? Content should be developed independently from system implementations, again to identify issues in interpretation and assumptions.
- Has there been demonstrated interoperability, i.e., do the different content examples work with the different implementations?

Once the specification is in place and demonstrated to work, several steps need to be completed to add the specification to the SCORM bookshelf.

- Determine additional criteria and changes to the overall SCORM model needed to fully support the addition.
- Develop the profile of the specification. Clarify issues of interpretation. Identify and correct deficiencies. Select which features are to be included or excluded from the profile. Determine extensions.
- Determine modifications to the other books of SCORM to accommodate the addition.
- Develop appropriate sample implementations and content.
- Develop additional test suites and conformance criteria.

**A SCORM Roadmap**

The feature set and components of SCORM 1.3 are considered to be fixed and final. All of the items described above represent potential changes beyond SCORM 1.3.

This report does not propose any roadmap or work plan for the evolution SCORM, and does not...
establish or recommend any priorities for the evolution of SCORM. There is a variety of work to include all of the potential extensions outlined above, including: gathering requirements, evaluating existing solutions, developing solutions, producing proof-of-concept systems, specification development, and specification profiling, building upon established practice. There are numerous contingencies to be addressed and many dependencies and potential conflicts that need to be resolved to address the issues.

In addition, all work must recognize the needs of the community and the need to maintain SCORM 1.3 as a stable specification to permit vendor adoption, content development and feedback.

The reader is again cautioned: **this report does not imply or indicate what will happen to SCORM.**

### SCORM Issue Summary

The table below summarizes the aspects of each of the issues. Items (C--Complexity, E--Effort, I--Impact, S--Scope, R--Research, P--Practice, S--Standards) are categorized on a scale of 1 to 5 (from low/easy to high/difficult), 0 indicates not applicable, V indicates variable (based on subissues).

<table>
<thead>
<tr>
<th>Issue</th>
<th>C</th>
<th>E</th>
<th>I</th>
<th>S</th>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Extensions to SCORM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presentation</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Navigation</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Assessments</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Simulation Content</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Repositories</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Data Models</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Learner Profiles</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Security</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Communication Models</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Web Services</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Collaboration</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Service-Based Content</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Localization</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Internationalization</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Accessibility</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Content Variants</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Digital Rights Management</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Category</td>
<td>Score 1</td>
<td>Score 2</td>
<td>Score 3</td>
<td>Score 4</td>
<td>Score 5</td>
<td>Score 6</td>
<td>Score 7</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>Competency-Based Learning</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Performance Support</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Mobile Devices and Mobile Learning</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Adaptive Systems</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Intelligent Tutoring Systems</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Simulation Systems</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Gaming Systems</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Enterprise Integration</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Infrastructure Extensions to SCORM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic Web</td>
<td>5+</td>
<td>5+</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Versioning</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Migration</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Specification Harmonization</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Emerging Specifications</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Specifications into Standards</td>
<td>2</td>
<td>V</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>