

EXECUTIVE SUMMARY

SCORM is the de facto industry standard for e-learning interoperability. There can be no doubt that it has been a tremendous success. SCORM has reshaped the elearning industry into a network of independent vendors capable of interoperating with one another through a common set of specifications.

A large part of SCORM's success is due to the interoperability created by its powerful yet simple communication framework. The SCORM Run-Time API is an elegant solution to a complex problem. It is relatively simple to implement once understood and it appropriately allocates complexity to the systems best equipped to handle that complexity (the LMS). This communication framework was visionary for its time, however it is beginning to show its age.

SCORM is now 10 years old, an eternity in the world of rapidly advancing technology. Over the last decade, two things have happened. First, as the adoption of SCORM soared, people wanted to use it in new ways, ways that stretched the limits of what SCORM was designed to do. Second, technology has continued to evolve at a breakneck pace. In comparison with today's technologies and best practices the current SCORM structure is antiquated and inadequate. Both of these trends beget the need for a more modern and powerful communication framework.

This project seeks to modernize the SCORM communication framework.

The project will begin by collecting, surveying and organizing use cases for the next generation of SCORM. We will quantify the shortcomings of the current ECMAScript API in relation to the requirements of the industry. Next, a survey of modern communication techniques will identify current best practices that are mature enough for inclusion in an industry standard. These techniques will be applied to an elearning context and matched with requirements to construct a communication framework that enables a broad spectrum of new capabilities. Finally, several of these techniques will be prototyped to prove and assess their viability.

The end result of this project should create a clear path forward for the modernization of the SCORM communication framework. Ideally, this work will feed directly into the development of the next generation of SCORM.

TECHNICAL APPROACH

There are assumptions inherent in the current SCORM communication protocol that have served the industry well, but which are no longer valid for modern technology and training programs. These assumptions constrain what can be done in a standardized manner. Standardization and its resulting interoperability are enormous catalysts in driving industry progress. For e-learning to reach its next level of maturity, these assumptions must be revisited and the resultant constraints removed.

Some of the faulty assumptions inherent in SCORM include:

- Online training is done in a web browser
- Training is done on a PC
- Training is done in a connected environment
- Training is done by a single learner in isolation
- Communication is one way
- Content and LMS's live in the same place
- Training happens in an LMS
- Content is formally created and needs to be durable
- Instructors don't interact with learners in self-paced modules
- The context and learner's history is only slightly relevant
- Security isn't an important concern

These assumptions are holding back progress and innovation.

Trainers want to be able to leverage increasingly available and affordable simulations.

Learners want to take training when and where they want to, be it on the PC on their desk or on their mobile phone on an airplane.

We live in an increasingly connected and co-dependent world. Working in large and diverse teams is the norm. Training should reflect this reality and allow us to develop the competence of a team jointly and as a whole.

Broadband is becoming ubiquitous and cloud computing is rapidly increasing. These days our data and our systems simply exist in the cloud. Our training data and systems should be no different.

In the information age, learning is a part of our daily lives. Learning isn't a process that can be prescribed and controlled. Learning can happen at any place and at any time. If there is a need to document learning events, it needs to be possible outside the context of a formal LMS. The web has transitioned from a read-only system to a read-write system. The best content isn't always curated or formally developed, it is user generated and potentially short-lived. Such content has vastly different requirements than formal online training programs.

These requirements are all very real and very essential to the modernization of SCORM. In an agile and connected technology world, the limitations imposed by SCORM are simply unacceptable. For SCORM to remain relevant and the de facto industry standard, it must keep pace with modern technology.



A proper solution to this problem must consider an exhaustive list of requirements and use cases. These must be assembled, organized and prioritized.

Our project will begin with this assembly of requirements. We will rely on three primary sources for requirements.

The first and primary source of requirements will be ADL. ADL has been listening to community requirements about SCORM for years and has assembled a significant list of requirements from its various constituents.

The second source of requirements will be the community white papers submitted to LETSI as part of the initial SCORM 2.0 effort. This repository of over 100 white papers contains a wealth of information about what the community needs SCORM to be able to do. This repository will be surveyed and cataloged.

The third source of requirements will be Rustici Software's client base and product users. We are uniquely positioned at the center of the SCORM world and see how SCORM is used in the real world on a daily basis. Over the years, we have collected a number of requests and shortcomings of the specifications. We will tap this vast resource and issue a survey to our clients in search of additional requirements.

Once the requirements are gathered, the next step is to prioritize the requests. Rustici Software will suggest an order of prioritization based on our experience, judgment and industry knowledge. However, ADL will play a pivotal role in guiding the direction of the project at this phase. Rustici Software's participation in IPT meetings will allow coordination with the rest of the ADL technical team to ensure that our efforts are in line with the strategy and direction of the ADL.



In Phase 2, we will look at the current state of software and web development to identify communication technologies, specifications and best practices that could be applied in the training realm to fulfill the identified requirements. We will analyze each technology to determine its feasibility for inclusion in the next generation of SCORM. The feasibility will be determined based on the technology's:

- Current level of adoption
- Technical maturity/stability
- Robustness,
- Support in major platforms
- Ease of use by developers

Communication tools will then be matched up with requirements. For instance, SOAP web services communication might alleviate security concerns. Bookmarklets might offer an appropriate solution for tracking informal learning. HTML 5 websockets might provide an effective way for learners to communicate with fellow team members. Comet technologies might offer a way for instructors to communicate with learners.



In Phase 3, we will begin to determine the most efficient and powerful path forward. Once we quantify the requirements that a particular technology can address, it becomes straightforward to identify the communication technologies most worth pursuing. Promising solutions will be chosen based on their applicability to high priority requirements as well as their ability to fulfill many requirements simultaneously.

Our current analysis is that modern communication is powerful, yet lightweight and focused on ease of use by its clients (much as SCORM has allocated the burden of complexity to LMS developers to pave the way for content developers). One way that modern systems achieve ease of use is by providing a flexible framework that adapts to the environment in which it is being used. Adaptation occurs in several dimensions, including level of complexity, software platform, and communication data formats.

A great example of a modern communication framework is the Flickr API (<http://www.flickr.com/services/api/>). Flickr defines a common set of concepts and core functionality, but then it provides a myriad of methods to access this core functionality. Let's take a look at the adaptations Flickr makes to accommodate its users:

- **Level of complexity:** Flickr provides a full specification for how to interact with its full set of functionality at the HTTP transmission level. If you want to get down and dirty and implement the most advanced functionality, you can do that. But, if all you need to do is make a simple picture upload from C#, you can just download the C# library and make a single function call. The complexity can be scaled up or down to match the utility of the service.
- **Software platform:** Flickr provides 15 client libraries each abstract the complexities of the underlying model and allow developers to work in their native language with paradigms consistent with that platform.
- **Data formats:** Even within the core API, Flickr allows you to choose how you want to send and receive data. Developers can choose to work with REST, XML, SOAP, JSON and even PHP data formats.

Flickr doesn't provide a communication API, it provides a communication framework with many different APIs that developer's can choose amongst. SCORM can learn a lot from Flickr and other similar communication frameworks.

The current SCORM API provides a great solution to a single problem. However, a single API is likely insufficient for solving additional problems. This project focuses on the construction of a communication framework, encapsulating multiple modalities to create a comprehensive and powerful solution to e-learning interoperability.

In any system, there is an inherent tension between power and simplicity. Modern communication frameworks seek to balance the two by providing power to those who need advanced functionality and simplicity to those who don't.

SCORM is no different. Requirements for improvements to SCORM are equally split between those that look for advanced functionality and those that just want something that is simple and which works.

Phase 3 includes the construction of a conceptual framework that embraces this tension. We expect to deliver a high level, yet operable framework for e-learning communication that accommodates both highly technical advanced use cases and simple data exchange.

To vet the framework, we will develop working (yet limited functionality) prototypes. These prototypes will be implemented on modern and commonly used software platforms and will seek to demonstrate the feasibility of the communication framework.

INFO



ABOUT RUSTICI SOFTWARE

Rustici Software creates products that simplify conformance with learning standards like SCORM and AICC. While other companies tolerate them, we embrace the standards and their intent. Really, *this is all we do*. Read more about it on [our website](#) or subscribe to [our blog](#).

COPYRIGHT INFO

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License. To view a copy of this license, visit [Creative Commons](#) or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

WHAT YOU CAN DO

You are given the unlimited right to print this manifesto and to distribute it electronically (via email, your website, or any other means). You can print out pages and put them in your waiting room. You can transcribe the author's words onto the sidewalk, or you can hand out copies to everyone you meet. You may not alter this paper in any way, though, and you may not charge for it.