White Paper

Domain enhanced sequencing and scenario generation in a web based environment

LETSI White Paper Solicitation on SCORM 2.0

Ohad Bukai, Jacqueline A. Haynes, Thomas Wason, Bob Pokorny

Intelligent Automation, Inc.

15400 Calhoun Dr. #400

Rockville, MD 20885

obukai@i-a-i.com

Abstract

In many professional domains, there is an interest in creating recurring training scenarios that reflect specific procedural tasks or protocols that a practitioner should follow within that domain, as part of a *learning by doing* instructional methodology. Examples of domains that regularly benefit from such training are aircraft technical maintenance, and medical practitioners learning diagnostic or surgical methods. Using current SCORM sequencing mechanisms, it is extremely labor intensive to implement the sequencing code for training that will follow this methodology by requiring the learner to follow a simulated procedure in a training environment; indeed, most learning by doing scenario trainers are implemented using proprietary non-SCORM conforming solutions. In this paper we describe STENV (Scenario based Training Environment), a method and approach that will enable training content developers to build trainers for procedural tasks, leveraging on the availability of standards to reduce development complexity, while supporting the designer with a sound tested instructional framework guiding the development task.

Problem definition

Some researchers have questioned the instructional value of the model on which SCORM is based. The focus of the critics was whether atomic de-contextualized learning objects can support a realistic pedagogic goal and whether it could convey a unified learning experience. A response to these concerns was introduced to the SCORM community through the introduction of IMS simple sequencing, which, when used correctly can put order and make pedagogical sense out of singular learning objects. Unfortunately, IMS simple sequencing does not make the instructional designer's task of bringing instructional order and meaning through sequencing easy. The standard is highly technical, highly procedural, and lacks support for associating instructional meaning with the sequencing code. We point to two types of such associations that are missing: *instructional* segments and *domain* related segments. IMS attempts to address this issue by supplementing its simple sequencing standard with the IMS Learning Design standard [3] , but this standard provides templates for learning in a multi-person/entity setting and is not really applicable to the web-based, single learner setting which is the focus of SCORM 2004.

Our experience in many web based training initiatives we are involved with reveals some common themes that , if better addressed in SCORM, we believe would promote its adoption . These include the following requirements:

- support for an evolving scenario
- multiple parallel "portals" (or SCOS) to interact with the student
- a formal user interface definition, that is external to the SCO
- modeling a wide variety of pedagogical structures
- modeling domain-specific segments and protocols

This paper addresses both pedagogical and technical issues. Although we describe concepts that go beyond traditional SCORM-like content development, our design also supports the traditional content development methods and products.

Demonstration use case

In our evaluation of the problem we explore the following use case:

An instructional design organization is tasked to create a SCORM-compliant aviation technical maintenance trainer, in which the user has to follow a procedural protocol for troubleshooting, using a troubleshooting console. The trainee is given a scenario with which he responds in a step- by-step manner. The same troubleshooting protocol can be repeated with multiple scenarios. We purposely select this use case as it demonstrates some of the challenges a designer has to overcome while developing a training product that is more complex than a page turner. Especially we look at the following features which SCORM currently does not support: (1) Implementation of an experiential instructional design concept (learning by doing). (2) Reuse of repeatable domain specific procedures (troubleshooting in this case) or segments thereof as part of the standard. (3) Definition of a user interface that is reused with different scenarios (the troubleshooter console in this example) (4) Parallel activities or channels of information (a coach, reference documentation etc) in SCORM. (5) The instructional design process is being integrated into the standard.

Many of the above items will have to be developed by the developer from scratch, even though they are heavily used by the community and could be reused. The need to develop these above capabilitities from scratch may be a reason not to develop the project to conform to SCORM, or to address SCORM in a shallow manner (e.g. as a single SCO).

The librarian metaphor

There are two types of librarians. Librarian type A is very neat; she records each item with its bibliographic information, and can quote metadata information about anything from title of a book to blood type of the author. This detail is her passion. The Type B librarian on the other hand gets excited when many of the books are actually in circulation, even if some of them cannot be found or end up being shelved incorrectly

Figure 1a: Type A librarian

Figure 1b: type B librarian (All books are in circulation)

This fictional classification of librarians is presented to distinguish between two competing approaches for standards for online learning. There is a trend that traceable to the early days of SCORM that prefers the record keeping functionality over usability, and specifically over giving instructional designers methods and processes that reduce their workload rather than overwhelm them. Initiatives that try to build bridges between SCORM and DITA / S1000D for example, focus on recordkeeping rather than ease of development, and instructional design support. Without discounting completely the value of record keeping, we suggest here an approach that focuses more on the relationship between standard support to instructional design product development, and the overall quality of the end product.

The significance of sequencing

We refer to sequencing as a set of instructions that describe how content is delivered in run time; it describes both the ordering of content delivery, and branching that is responsive to student's interactions. Although sequencing has been used in web based training for a while now, the industry has not yet embraced the potentially significant role of sequencing for the quality of the product. The problem is that currently sequencing is abstract, difficult and time-consuming. Sequencing code was traditionally used to *hardcode* behaviors that reflect some planning. What the industry failed to acknowledge is that sequencing code can be *reused*, enabling us to define sequencing *segments* that have some normative meaning (*instructional* or domain *related*) and reuse these segments or their variations in different contexts, or different scenarios, thereby increasing the functionality while reducing the cost of developing and using such segments.

3 conceptual layers for sequencing

In instruction, sequencing mechanisms perform three major tasks:

Technical control of activities(TCA)– This aspect addresses the traditional capability to provide technical control over flow, creating the capacity to link content with activities, read and write into variables to maintain persistence, affect the flow of activities according to values of variable, etc.. This aspect is merely a control mechanism, and as such it does not have *normative* "knowledge" about the meaning of the flow it controls.

Instructional control of activities (ICA) – In practice, well tailored sequencing structures represent instructional strategies with normative pedagogical meaning. Instructional designers can use TCA to create sequencing segments that have instructional meaning (ICA), and reuse them while maintaining their instructional meaning. Simple examples are pre-tests, post-tests, remediation, drill-and-practice, and so on. Potentially, sequencing segments can be created using the technical control flow mechanism, to address normative instructional objectives, and then be reused as such.

Domain specific control of activities(DCA) – Well tailored sequencing structures can also mimic procedural tasks / protocols that are important in a specific domain. Such structures can be than reused in different scenarios and use cases that follow the same protocol. In domains where procedural tasks are central to the training of the practitioner, many times learning by doing is the instructional method of choice. Here, TCA can be used to create sequencing activities that replicate protocols used in a specific domain. Repeatable maintenance diagnostics or medical diagnostics are examples. In both cases, the protocol stays constant, while the details of each case change.

In the proposed solution we address each layer as a separate *abstraction,* or a logical set of specifications that addresses the specific layer implementation and use, with the intention that, by using these specifications the users (developers, instructional designers) will have capabilities to address complex concepts without the need to understand or manage their internal structures.

Current web based sequencing standards (SCORM/IMS), address only TCA. When content developers wants to implement ICA or DCA, they have to develop their internal logic in SCORM, or as it happens in many cases, create a non-SCORM training system, or adhere to SCORM on a surface level only. It is our intention to describe a framework for a SCORM 2.0 that will address TCA, ICA, and DCA.

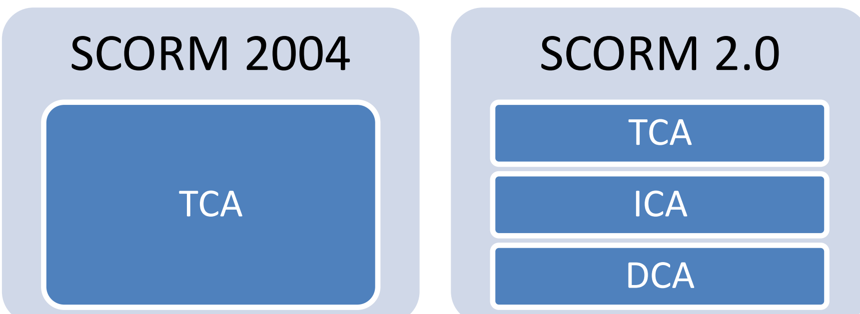

Figure 2: 3 conceptual layers of sequencing

Hierarchal relationships of the conceptual layers

This separation of sequencing functionality into layers is done in the interest (discussed later) of creating abstractions that will be detailed by a standard and handled by an automated editor and run time environment. To this end we wish to address the hierarchal relationship between the three layers. This way developers of instructional activities, while authoring, will be able to leverage the functionalities of the underlying abstractions while focusing on their task (domain related, instructional design, content scenario creation) paying less attention to developing the required infrastructure, modeling, or other sub tasks that are not directly related. We evaluated several topologies for the relationships between the layers and found that the, Figure 3 topology to be most appropriate.

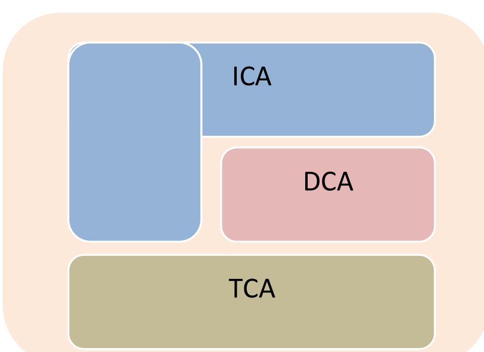

Figure 3: Topology for the conceptual layers

As described in Figure 3, the underlying layer is the TCA. It provides the basic sequencing capability. Traditional training projects that do not need advanced modeling, can limit their use at this layer as it is comparable to IMS simple sequencing in its functionality. DCA uses the sequencing functionality that TCA provides to enable developers to create sequencing structures that replicate protocols, segments, and processes that are used in that domain (e.g. troubleshooting protocol for maintainers), therefore DCA is logically positioned on top of TCA. ICA, on the other hand, utilizes both TCA and DCA: it utilizes TCA to create reusable instructional segments, and it utilizes DCA to properly sequence domain related segments within an instructional context.

Dedicated reusable UI components

Traditionally, user interface has been treated as *content* in the context of standards. SCORM, for example, does not address at all what a user interface *inside* a SCO looks like; IMS sequencing is also silent about user interface. In practice, the user interface is more central in relation to sequencing, especially in cases when the same user interface is used to control the flow of a scenario, or is reused across scenarios. In these cases the definition of the user interface should be elevated above the content to a function that controls the flow. For example, if we use the troubleshooting scenario use case, the user uses a portal, and follows a known protocol to solve a troubleshooting scenario. Throughout the training, the user interface of the portal does not change, only the content that it presents changes. Moreover, the same user interface is reused with different scenarios. We assert that in this context UI definition is more sequencing rather than content, and using our topology, DCA should be able to have control over the definition of the interface, rather than traditional sequencing code.

Single Vs. Multiple portals of contact

Current web based instruction standards enforce a mutually exclusive control of one instruction unit at any point in time. In SCORM this means that only one SCO can run at a time. Contrary to that approach, many pedagogical approaches promote strategies that follow the student *across* activities and:

(1) Develop links across content

(2) Put the current learning object in a broader context

(3) Supply external references when the student needs them

(4) Provide coaching

(5) Address specific strengths and weaknesses of the student

(6) Provide continuation support for discrete items

(7) Link specific material to concepts (semantics).

We refer to these items as "portals of contact". These are channels that have access to the student (or the student has access to them), and their existence may have a different scope or time span than the specific current learning object. A variety of well-regarded pedagogical models support such external activities; Two examples are: (1) Micro-learning theory, where an external entity or activity would help the student contextualize the content, and (2) Cognitive Apprenticeship [4] where activities such as modeling, coaching, articulation, reflection, or exploration are part of the learning experience. . Although there is not final list of such 'portals', some that are well accepted are: (1) coaching (2) external references (3) technical documentation (4) conceptual mapping (5) help (6) collaboration / discussion. The need for such measures to support more complex learning environments suggests that the mode of one learning object presented at a time may be limiting, and should be replaced with *entities* that can run in parallel, and mitigate the gap between the short term objective of the learning object and contextual needs, student needs and objectives of a larger scope.

Stakeholders

Content development organizations – These are commercial / educational organizations that focus on the generation of content for education and training. They generally have commercial interest in reduction of cost, complexity, and overhead associated with content creation. They will generally enjoy lower costs, and will be able to deliver content for a lower price, should a standards effort reduce the effort of creating the content. They will also gain value from integration of standards with instructional design practices. In the vision of STENV, content development organizations will develop scenarios according to models described by professional communities.

Professional communities – In many professional domains, professional communities organize to create standards that address the various needs of their specific community. Repeating a previous example, the medical community addresses determining the correct protocol for medical diagnostics, the same as the aviation technical community formalizes the aircraft troubleshooting protocol. In the context of STENV, it is the responsibility (and interest) of a professional Community, to describe: (1) reusable sequencing templates that follow the protocol of the community. (2) requirements for user interface definition, for user interface generally used in their context' (3) Data Definition of what a model of a scenario looks like.

Tool developers – These organizations follow standards to develop authoring tools to help instructional designers author content that adheres to the standard, and to assist instructional designers in the in navigating the instructional design process.

LMS vendors – These organizations develop tools that deliver the training activities to the end user (student or trainee). They must implement standards to create server-based components that can deliver content that was conforms to the standard.

SCORM users – These government, educational and commercial organizations deliver training using SCORM compliant content and a conformant LMS. Their interest is in the balancing of new functionalities with backward / legacy compatibility.

LETSI- The above described STENV framework relies heavily on the participation of standards bodies to bring professional communities together for the creation and maintenance of specifications and standards, and to enable finding ways for reuse of standards' products across domains. It seems that LETSI is well positioned to play a central role as a forum that binds these communities together and promotes the use of such standards.

Proposed solution

The proposed solution entry point is current standards from which we diverge to address our goal, while respecting legacy needs. As the scope of the white paper suggests, we will discuss the proposed solution by highlighting its main components.

Modular sequencing language

In order for sequencing code to be modular, and to address the various aspects of processes in instruction (pedagogy, domain) it has to follow object oriented principles. This can be achieved by strong typing[1], encapsulation of behavior, and inheritance of sequencing code segments. Basic components of the proposed object oriented sequencing code are:

[1] A programming language characteristic that provides strict adherence to the rules of typing. Data of one type cannot be passed to a variable expecting data of a different type.

Primitive sequencing code – This is a procedural language that controls the flow of instruction and links to resources. IMS simple sequencing is an example of this type of code.

Strong typing of sequencing code – Primitive sequencing code can be annotated with a type name and as such reused to perform some defined repeatable behavior. Input parameters can augment instances of that code in a way that is pre-defined (For example, setting resource, scenario data, changing length). Potentially, a new type can be created also by combining primitive sequencing code with other pre-created types (Figure 4a). An instructional example would be a pretest / post test. In this type, a pretest is administered, then some learning occurs, and at the end there is a post test. The context may change (history or math), and modifiers may change (as the length of the learning) but the basic structure persists. Of course, this basic model can grow to more abstract levels to address segments such as "troubleshooting protocol" for example.

Sequencing content – This component represents a set of references to resources and initialization information that creates a *sequencing instance* out of the sequencing type. (Figure 4b). A sequencing instance (in instructional terms) can be a scenario.

Specialization across sequencing types can be addressed using polymorphism[2] by adding sequencing primitive code to a Sequencing type. (Figure 4c).

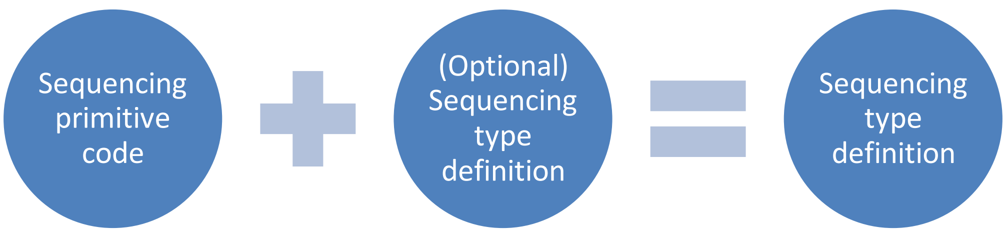

Figure 4a

Figure 4b

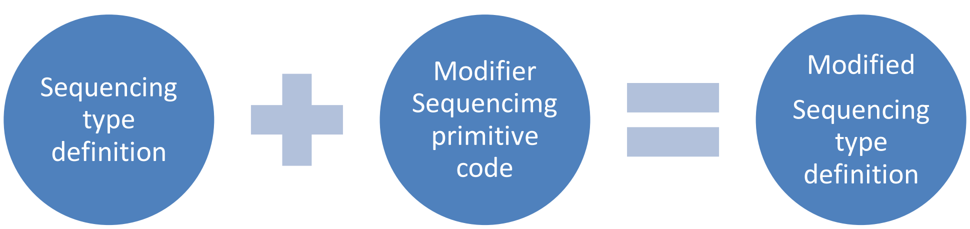

Figure 4c

[2] A method that allows values of different data types to be handled using a uniform interface

3 Layer sequencing modeling

The above description of a modular sequencing language lays the foundations for an implementation of the three layers described earlier. Sequencing primitive code will support the underlying TCA functionality, while the type definitions will enable users and communities to develop reusable sequencing structures to eventually establish pedagogical and domain specific segments (ICA and DCA respectively).

MVC methodology for DCA

MVC (Model View Controller) is an industry method for distinguishing between model, presentation and control units that translates a model into a presentation. Successful use of the pattern isolates business logic from user interface considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other. In MVC, the Model represents the information (data) of the application and the business rules used to manipulate the data; the View corresponds to elements of the user interface; , and the Controller manages details involving the communication between the model and user actions.

These three components are represented in STENV as:

Model – Represented by sequencing content (See above Modular sequencing language section).

View – Described by a user interface definition (See next section).

Controller – Sequencing code, represented by strong typing of segments.

User interface definition

STENV will incorporate a UI definition standard as part of DCA, to enable the use of reusable user interface components. A sequencing code will use (when applicable) the user interface definition to render the user interface for the user, and based on the protocol and scenario, supply input for the user interface to show. This will require the following components:

1. Formal definition of a user interface, its inputs and translation of user interaction into sequencing actions
2. Run time environment that can read the user interface definition, render it, and convert user interactions into sequencing actions.
3. Data model definition for scenario stage details, readable by the runtime environment to be presented by the user interface.
4. Sequencing code that incorporates selection and manipulation of user interface actions as part of its code.

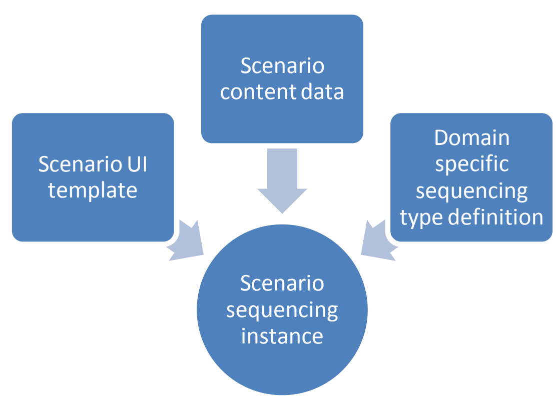

Figure 5: Scenario sequencing

Integration and other technical issues

Backward compatibility with SCORM 2004 content

Although implementation details are beyond the scope of this white paper, there is nothing in STENV that precludes incremental implementation that builds on, and extends SCORM 2004. For example, SCORM/ IMS sequencing can be the implementation of TCA, and current CAM model can support much of the requirements we describe. It seems that some leverage of SCORM 2004 can go a long way in mitigating the legacy content issue.

Involvement of standards committees and open specifications

The basic assumption of STENV is that through community collaboration and the work of standards committees, some procedural structures, both instructional and domain related can be defined and shared across the communities. Accordingly, in order to implement a standard in the spirit of STENV, community participation will be essential, as well as a description within the initial standard regarding how the process of defining new structures will be conducted. A good analogy would be the process of certifying a new XML schema with the right regulatory body. The wider community uses the same XML schema while sub communities use their own sub schemata.

Existing Implementations, Prototypes

Implementation of related concepts by Intelligent Automation Inc.

As of now, STENV is a design concept only. Some aspects of STENV are being implemented by Intelligent Automation Inc. (IAI), in a project named ASSIST. In ASSIST, SCORM sequencing in used in modular ways in order to create reusable instructional constructs. These constructs in turn will reduce the difficulty of creating complex instruction.

Example of projects developed by Intelligent Automation Inc. that would benefit from STENV

To demonstrate the relevance and importance of STENV we attach some screenshots of web based training IAI has been tasked to develop within the last few years. Many of the functionalities described above were used in these projects, having the proposed standard in place would have benefited them significantly, as much of the functionality described in this document had to be developed in-house, instead of leveraging from the availability of a standard. The common themes that these projects and STENV share are:

1. Underlying pedagogical model in sequencing.
2. Underlying sequencing model that is domain specific (Intelligence, ATC, Military tactics)
3. Reliance on a scenario.
4. Reusable segments and user interfaces across scenarios.
5. Multiple portals of learning with variable time frames

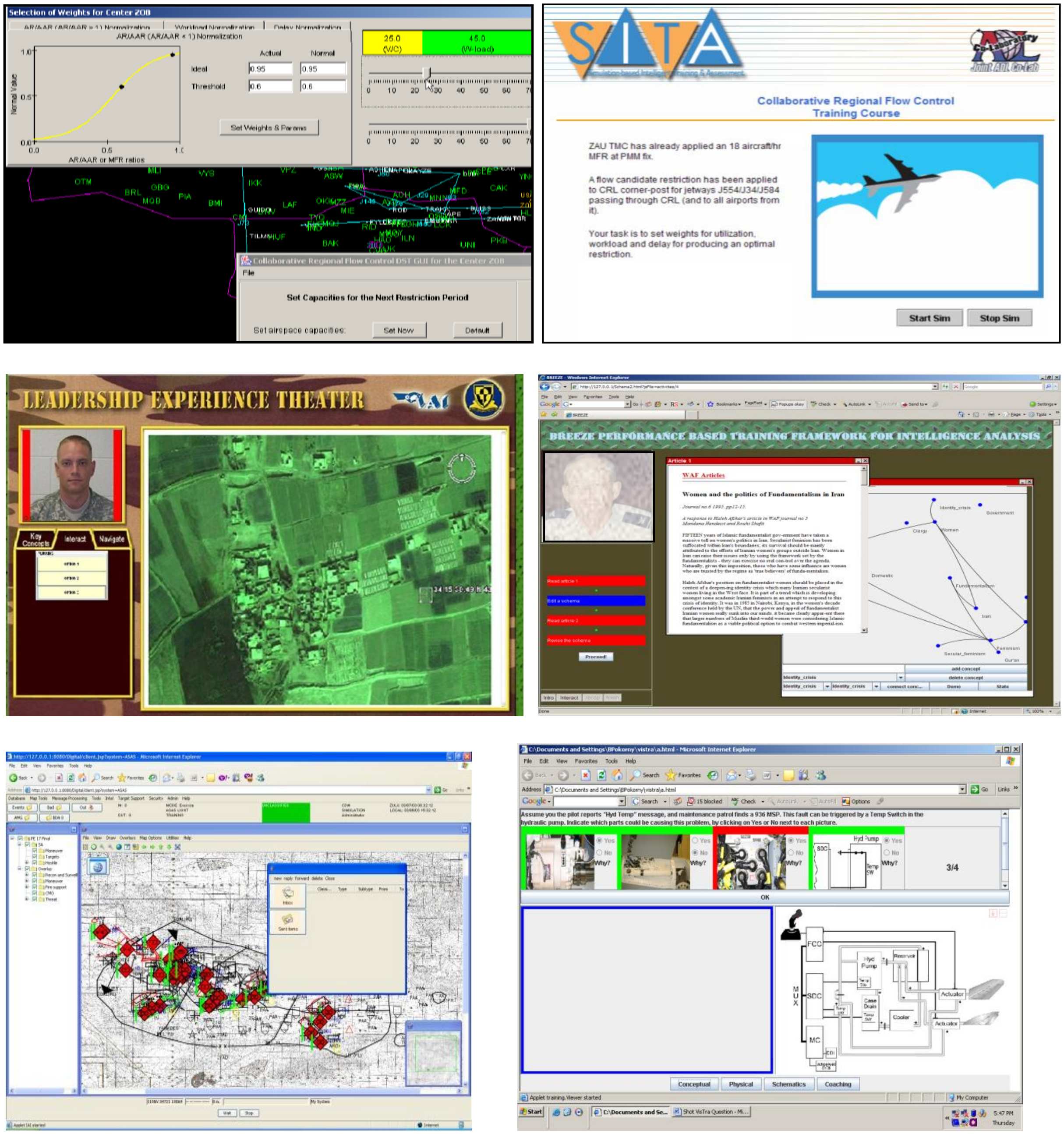

Figure 6: Examples from practice

Alternative approaches taken by others

There are several other approaches that address domain specific needs.

S1000D and SCORM -The upcoming S1000D v4 draft version has introduced the learning data module as a method to describe learning content within S1000D, control versioning and presentation. The focus here is in helping the organization maintain version control and other cataloguing requirements rather than address the learning experience. This solution is industry specific (aviation technical documentation) and may not apply or be generalized to other communities.

DITA and SCORM **-**Integration of DITA and SCORM has been discussed for a while [1]. The criticism regarding this discussion was mainly that while standard groups addressed mainly standardization of information modules andreuse of content in an instructional / documentation environment, they neglected to address the real problem of “how can the reusable objects be reconstructed in an instructionally valuable way”?

Summary and recommendation

STENV is a concept designed to support high level of functionality in scenario-based instruction that focuses on “learning by doing; ” as such, it extends the envelope of types of activities that can be delivered through “SCORM-like” standards, by creation of high level reusable constructs that focus on both instructional repeatable segments as well as domain specific reusable segments. To support this, STENV addresses the following

- Three layers of sequencing (Technical, Instructional, and Domain) and associated modeling.
- Support for parallel channels of instruction.
- User interface definition as part of the sequencing definition.
- Support for the process and development of sub sequencing standards to address needs of specific professional communities.

References

[1]DITA Proposed Feature #12058, http://www.oasis-open.org/committees/download.php/25901/Issueditalearningspec12058.html

[2] SCORM: clarity or calamity? SCORM is beginning to change the e-learning industry, but will it clarify and strengthen it or lead it into disaster? , 2002, Online Learning Magazine.

[3] http://www.imsglobal.org/learningdesign/

[4] Collins, A. Brown, J.S. and Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. American Educator, 15(3), 6-11,38-46.

Bio of Authors

Mr. Ohad Bukai has been a researcher at Intelligent Automation Inc for the last five years. After receiving his M.S. in Computer Science from Polytechnic University, New York, he worked on business intelligence applications, text based algorithms, and applied research for the development of instructional software and simulations.

Dr. Jacqueline Haynes is co-founder, Executive Vice President, and Director of the Education and Training Technology Group at IAI. Her background combines education and psychology with AI applications. She received her Ph.D. from the University of Maryland in Curriculum and Instruction. Her research interests include research-based instructional design, tools for Web-based instruction, and reading comprehension.

Dr. Thomas Wason is a senior scientist in Intelligent Automation Inc. He holds a Ph.D. in Experimental Psychology, North Carolina State University, Raleigh, NC (1993). He has been actively involved in various standards bodies. He was involved in Developed XML-Schemas for IMS specifications, and development of IEEE Standards for metadata, taxonomies, XML-Schema and e-learning architectures.

Dr. Bob Pokorny is a Senior Cognitive Scientist at Intelligent Automation, Inc. He, has conducted research and created training systems for over 20 years. He has designed and evaluated Intelligent Tutoring Systems for the Air Force. . He led computer-based training projects, including team training projects and an integration of S1000D and SCORM standards.