S1000D SCORM Test Bed: Integrated Development of Technical Publications and Training

Kevin Ruess, Bob Pokorny, Jackie Haynes
Intelligent Automation, Inc.
Rockville, Maryland
kruess@i-a-i.com, bpokorny@i-a-i.com, jhaynes@i-a-i.com

Tim Tate
Job Performance Technology Center
Alexandria, Virginia
tatet@adlnet.org

ABSTRACT

Operating and maintaining modern weapon systems requires vast amounts of information. Technical publications must provide maintainers the information they need for complex equipment. The use of this data is expanding with the concept of "net-centric warfare." The data is no longer paper; it is electronic and used in such things as simulations, 3D graphics, Electronic Performance Support Systems (EPSS), and games. These materials change frequently over the life-cycle of the weapon system, and each sample will have its own variants of a common function (e.g., one squadron will have an updated radar system, while another will have an older version). This technical documentation information is often represented within a Common Source Database (CSDB) in S1000D format, and current trends point toward wider adoption of S1000D.

The technical documentation has many functions within the information sources for modern equipment. The Office of the Secretary of Defense (OSD) funded the S1000D SCORM Test Bed project specifically to identify and evaluate methods for integrating S1000D-based technical manuals and SCORM-based training. The primary advantage of integrated S1000D and training development is that when technical documentation changes – and the technical publications and training content are explicitly linked – updating or changing specific sections of training content becomes much easier to manage.

To explore how the "trusted data" can be used for technical publications and training, we created a test bed with examples of integrated development of technical publications and training. We describe three different approaches to integrating technical documentation into training and the ways Program Managers should apply best practices for this integration into an example product.

ABOUT THE AUTHORS

Kevin Ruess is an instructional technologist with Intelligent Automation, Inc. and led the architecture team for the S1000D SCORM Test Bed Project.

Bob Pokorny is a cognitive scientist with Intelligent Automation, Inc. and served as Program Manager for the S1000D SCORM Test Bed Project.

Jackie Haynes is the founder and Executive Vice President of Intelligent Automation, Inc.

Tim Tate is of the Job Performance Technology Center at Advanced Distributed Learning (ADL). He promoted and provided oversight for the S1000D SCORM Test Bed Project.

S1000D SCORM Test Bed: Integrated Development of Technical Publications and Training

Kevin Ruess, Bob Pokorny, Jackie Haynes
Intelligent Automation, Inc.
Rockville, Maryland
kruess@i-a-i.com, bpokorny@i-a-i.com, jhaynes@i-a-i.com

Tim Tate
Job Performance Technology Center
Alexandria, Virginia
tatet@adlnet.org

THE WORLD OF TECH DATA AND TRAINING

Though technical publications and training present overlapping content, the two sources have traditionally been created and managed separately due to organizational boundaries. The technical publications and training in the military have recently adopted different specifications, Shareable Content Object Reference Model (SCORM) for training, and S1000D for technical publications. When equipment is changed, both the technical publications and the training have to be changed. If they are not integrated, changes increase the odds they will be out of synchronization. The integration of training and technical publications is desirable for several reasons: cost savings, timeliness, and improving warfighter readiness. When content is shared, changing the content once revises both technical publications and training, saving duplicate effort, and reducing cost.

The training community benefits by using technical publications to ensure relevance and integrity of products. The technical data community gains access to training information for completeness of manuals.

Increased Complexity and Frequent Change

Modern weapon systems are functionally sophisticated. The military needs to effectively train the users and maintainers of these systems. Increased concern for cost efficiency raises the stakes for drawing both technical data and related training materials from a single trusted source. Not only are systems more complex, they are more variable and can have many possible configurations. Each configuration, and the procedures for changing from one to another, increases the complexity of providing the right documentation and training at the right time.

Importance of Trusted Sources

Training materials must derive from authoritative sources. When the training involves the operation and maintenance of equipment, the trusted source is almost exclusively the approved technical manual. Using this source allows the instructional designer to focus on the task of creating effective learning while confidently relying on the accuracy of the source content.

The S1000D SCORM TEST BED USE CASES

Pressures for shorter production cycles and reduced lifecycle costs for both technical data and training mean that the two will have to be developed cooperatively and become more interdependent. These pressures provided both motivation and impetus for the Office of the Secretary of Defense (OSD) to fund the S1000D SCORM Test Bed. The technical documentation has many functions within the information sources for modern equipment. While addressing the challenges technically, the project has as its main audience the Program Managers. Their responsibilities now include acquiring technical data and training as efficiently as possible throughout the lifecycle of the weapon system.

We explored various approaches within the test bed. The use cases demonstrate the technical capability to integrate technical data and training, proving that they can be integrated. They also inform technical experts of some approaches and methods they can use. Working through these examples led us to describe Lessons Learned and Best Practices.

To demonstrate the Use Cases, we use the context of a "Tactical Bicycle" system to allow using S1000D data modules (DM) with no privacy or security concerns.

Integrating S1000D Content in Training

Money and resources of large products or systems (e.g., Joint Strike Fighter) can be saved by integrating technical documents and training. The advantages include sharing of common content, reducing needless duplication, and automatically incorporating updates to technical publications training. In this Use Case, we explore how technical documentation in S1000D format can be embedded in SCORM-based training.

In this Use Case we sought a standard, repeatable process by which SCORM shareable content objects (SCOs) could present content from an S1000D DM, common content shared, and updates to technical

publications easily incorporated into training. We considered two approaches:

a) Transform an entire DM into a SCO: This maintains the integrity of the DM. In S1000D, the control authority authorizes a complete DM. Some vendors provide solutions in which entire DMs are transformed through the use of templates into training.

b) Embed a subsection (<para>) of a DM into a SCO: While an entire DM may often be presented as a SCO, an instructional designer will sometimes wish to create SCOs that use a different division of content than was applied when creating the DM.

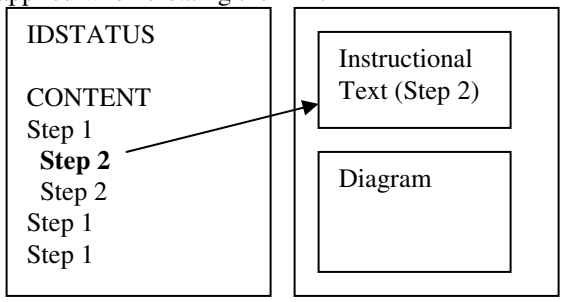

Figure 1. S1000D content embedded in a SCO

This approach, shown in Figure 1, identifies sections of an S1000D DM to include in the training; when the S1000D DM changes the training is updated. Here, we assumed a "black box" control authority for the sake of simplicity. Essentially, this solution links a subsection of S1000D content to a specific part of training as a general solution—it allows the Instructional Designer complete control over the division of the content, including the use of an entire, existing DM as a SCO.

To create this method, we needed to identify S1000D segments to be used in training and incorporate identified segments into the training in the SCO.

The method used two technical mechanisms: (a) we identified specific <para>s in S1000D for use in training by labeling them with the <id> tag (Figure 2, top) and, (b) in the SCOs, we incorporated specified sections from DMs into training by inserting an iframe, a rectangular area within a web page that sources its content from an external source (Figure 2, bottom). In this example, content is created by passing the DMC (data module code) and paragraph ID to an S1000D-to-HTML server applet (used for presenting the specified section of S1000D in a web browser).

To create this link, we built a tool to facilitate the insertion of <id> tags in S1000D DMs and the insertion of iframes into SCOs. This tool allows someone (most likely an instructional designer working with a technical writer) to view the S1000D content, review the <para>s, and then identify specific <para>s that should be used in training. Then the tool supports the insertion of the appropriate iframe construction into the HTML SCO.

```
<mainfunc>
  <step1 id="id 1110112992978A150">
    <para> Use the tire pressure gauge to check the tire pressure. </para>
  </step1>
  <step1> id="id 11084129929788151">
    <para> Take appropriate action depending on the tire pressure.  Inflate or
deflate the tire until it is 30 p.s.i. </para>
  </step1>
</mainfunc>
<closereqs id:"id_1148412992978A152">
  <reqconds>
    <noconds/>
```

```
        </div>
        <div id:"1id_114041299297BA152" style:"display:none">
http://127.0.0.1:8080/S1000D/jsp/files/DMC-BICYCLEXXXAAA-AAA-DAC-20-0000-400A-A_000-
005.xml?id=id_114041299297BA152<br><br>
             &lt/iframe   id='myframe'
Src="/S1000D/jsp/files/DMC-BICYCLEXXXAAA-AAA-DAG-20-0000-100A_000-
05.xml"id=id_1148412992978A152
Text=true" scrolling=-"nop" marginwidth="0" marginheight="0" frameborder="0"
vspace="0" style=overflow:visible; width:100";
Display:none" &gt &lt/iframe&gt/<BR>
         <input type="BUTTOM" value="back" onClick="show('id_114841299298A151')":
       </div>
    </body><html>
```

Figure 2. S1000D <id> tag and SCORM iframe

To summarize, this is one method by which S1000D technical data content can be inserted into SCORM training. When the DM is changed, a new version is created. To be presented over the S1000D viewer, it must be authorized by the control authority.

Equipment-Version Specificity for Training

In this Use Case, we explored how training could be tailored to a particular version of equipment to provide training that exactly matches the equipment the learner will use. We examined how equipment-specific training can be designed and presented using SCORM to call specific SCOs based on the equipment that the learner has. The SCOs that present equipment specific content could have embedded training content taken directly from S1000D DMs.

The equipment varies between learners, based on where the learner is, what the learner has access to, and possibly what the learner has been certified to do. The equipment configuration can be generated based on the learner's profile. The training a learner sees needs to be linked to the equipment configuration. The equipment configuration is assumed to vary from learner to learner, and so is stored as part of the learner's profile. Each profile has its own matching training.

We used the approach outlined below (which takes advantage of capabilities of SCORM 2004) to implement presentation of training that ties a specific learner to a specific equipment configuration:

1. Solicit the user's equipment configuration, and store as a variable (using global objectives for SCORM sequencing) in the LMS.
2. Write separate training content for each equipment type.
3. Write SCORM Sequencing rules to select the appropriate SCO(s) using global objectives to represent the equipment configuration.
4. Use a SCORM 2004 compliant LMS to present the right training for the learner.

Implementing this approach posed a number of technical challenges. The solution had to:

1. Identify equipment-specific factors (and their values) .
2. Create versions of instructional content that depend upon one or more of these factors.
3. Create, or provide a means to create, user profiles across these factors.
4. Create a way to set global objectives in the LMS based on the user profile.
5. Create sequencing rules that utilize global objectives to select appropriate content.

Creating customized training involves first identifying content (e.g. procedural training) that varies according to the defined factors. This content should be isolated so that the remaining content can be kept common. The instructional designer then creates versions for each value of the factor. Each variant is packaged as a distinct SCO. These SCOs, together with common content, are placed in a SCORM Content Package, adhering to the SCORM Content Aggregation Model.

We created a user profile by asking what type of bicycle the user has. In a production setting, profiles are likely to be inserted into the LMS automatically or manually entered by an administrator (as opposed to the learner). The response is encoded as a global objective, with a "satisfaction status" value "true" (for quick release hub) or "false" (standard hub); if there were more options for this variable, then a different encoding would be needed. We also created a web service to transmit the value to the LMS.

SCORM Sequencing rules are used to sequence content (SCOs) into a complete instructional experience. These rules can be used to select and present the correct SCO, according to the global objectives. For each instructional section with variant content we wrote branching rules as in Figure 3.

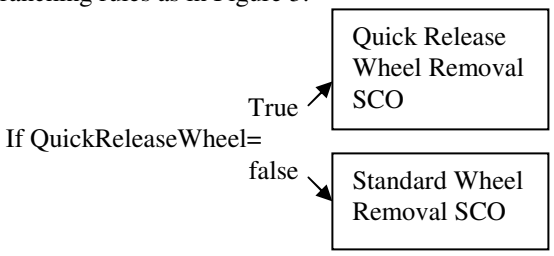

Figure 3. SCORM sequencing branching rule

The code implements the rule in the content package's manifest file. The simple sequencing rules specify which SCOs are presented to the learner based on the value of the global objective and the rules. A SCORM 2004 LMS will parse these rules to present the correct content.

SCORM sequencing rules can provide instructional synchronization between training and equipment To generalize from this Use Case, the equipment configuration can be represented explicitly with factors and values. These factors and values are represented using SCORM global objectives and simple sequencing rules to select and present the correct SCOs during training.

This approach focused upon delivering training selectively using SCORM; S1000D has the capability to present only the procedures that pertain (or are "applicable") to the particular equipment configuration using the <applic> tag. This Use Case demonstrated the capability of SCORM simple sequencing rules to direct training much like the <applic> tags direct IETM presentation in technical publications; real production work would use better tools to link training and technical publications

Tools are needed to facilitate branching in training that is already specified in technical publications. Of course, the branching required in training may be different than that used in technical publications, but the instructional designer should be aware of the branching in the technical publications, and then accommodate that in the training.

To use conditional statements in SCORM to adjust to equipment specific configurations, a production level support tool should be developed that can:

- Interpret the S1000D and identify equipment variables for which training should be equipment specific.
- Enable visualization of the branching based on equipment factors and include the factors (e.g., "wheel attachment to frame"), the levels of the factors (e.g., quick release versus standard release), and the SCOs to present for each case.
- Easily translate the described layout to conditional statements that execute in the SCORM Run Time Environment.

Proto-Content Trusted Source Approach

In the S1000D-SCORM test bed, we also explored how approaches that we applied to technical data and training may generalize to more types of content, including logistics databases, personnel management systems, and engineering assets and drawings.

The trusted source will not always be the technical publications, and the variety of authoritative sources available from the trusted repositories will continue to grow. These sources will include general and purpose-built repositories, databases, digital libraries, content collections, and other sources both public and private. Many sources of information may have trusted status for authors to use in developing derivative content for their end users. This Use Case explores some of the implications of defining and using proto-content as trusted sources for developing *derivative content*. Determining content that is trusted and content that is derivative depends on an organization's perspective.

Proto-content functions as the raw materials (or raw digital assets) which authors of derivative content use in the creation process. Proto-content in the collection is identified as trusted by the relevant authority and is available for use throughout the organization.

Proto-content includes but is not limited to: logos, legal disclaimers, laws and regulations, definitions, illustrations, lists of standard values, design drawings, design data, product breakdown structure, parts, FMEA/FMECA, maintenance task analysis, operator task analysis, LSAR, and safety analysis.

A collection of proto-content could be relatively static or quite dynamic. If it covers a domain unlikely to change (e.g., the laws of physics) then the collection would be quite static. If it changes frequently (e.g., guidelines for dealing with IEDs in battle areas) then the collection might change daily.

Derivative content depends in some way on some proto-content, but derivative content can include much more detail and is targeted toward a particular context or application. It is intended to be published, and is subject to business rules for approval, storage, registration, and dissemination.

Derivative content might include: a business card with a company logo, technical manual content produced from engineering documents which the writer "translates" into technical manual content for a particular audience, training materials an instructional designer produces from technical documents, product illustrations, marketing materials explaining which configurations of a product are available in which countries, etc.

We identified three types of users (an individual might be in both author categories):

1. Authors of proto-content
2. Authors of the derivative content (instructional designers, technical writers, etc.)
3. End users

Some common, general characteristics of proto-content include:

1. It is stored in its high-resolution form that meets requirements of authoring tools, using standard, open source data formats.
2. It is stored with relevant metadata (e.g., paragraph numbers in the tax code)

3. It is of atomic grain size (granularity to the point where further subdivision offer no benefit)
4. It is context independent (can be reused in many types of derivative content
5. It conforms to the requirements of the organization's trust model

Variations among characteristics of proto-content may cause issues when using that data as the trusted source for derivative content, when the aim of the derivative content is for it to be in an open, neutral, standards-based form such as S1000D. The choice of proto-content data formats should be made early in the project.

The approach rests on a basic architectural concept. The proto-content is stored in a collection. Authors use proto-content in products they develop, but those products (in this case SCOs and S1000D DMs) are stored separately in an S1000D Common Source Database (CSDB) or a SCORM repository.

Conceptually there is only one proto-content collection, though it might be geographically distributed and managed with a combination of solutions. There can be any number of derivative databases; they are not limited to SCORM and S1000D and could include databases for on-line help, marketing and sales materials, etc. Different repositories/databases can only obtain proto-content from the proto-content collection, not from each other (Figure 4).

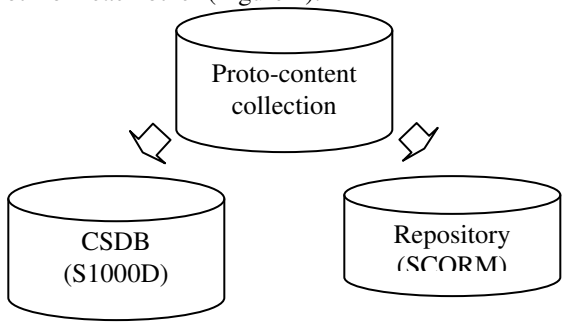

Figure 4. Proto-content architecture

All content subject to "reuse" must be trusted and must be derived from the proto-content in the proto-content collection; there is no flow or reuse between any of the derived content databases or repositories.

Organizations may develop rules for promoting content from status as derivative content to proto-content. For example, an instructional designer might develop a simple animation of the controls of a piece of equipment, and that animation might later be useful for a marketing presentation or as a demo in on-line help.

The types and nature of proto-content should be identified early in the project. Business rules will identify the authority responsible for deciding how proto-content is determined to be trusted, how it is certified as technically accurate and safe to use, and how it is determined to be fit for its intended purpose. Some proto-content will likely be sourced "as is" whereas much will have to be developed from scratch.

Proto-content objects are like ingredients that can be used in different recipes. In some situations, a proto-content object may be suitable for consumption by the intended audience without being combined with other objects or going through a preparation process; in such a case the content would be replicated "as is" in the repository according to project business rules (note that end users never access proto-content from the proto-database). A more likely scenario is that the author develops context-specific (non-proto) content during the authoring process that includes proto-content

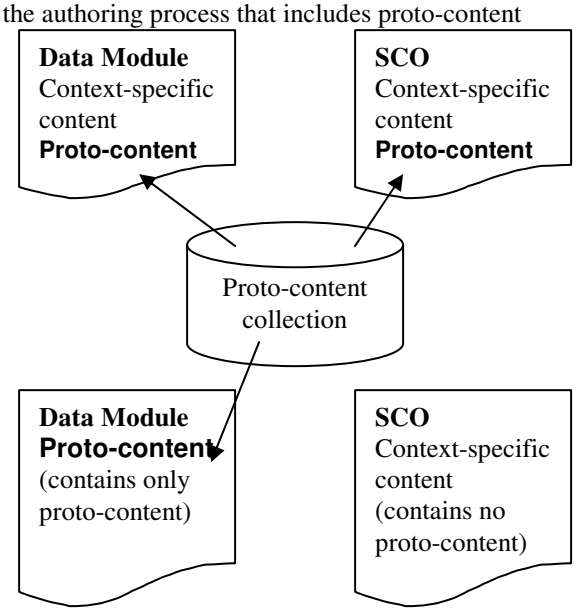

Figure 5. Proto-content and derivative content

As shown in Figure 5, depending on the requirements of the DMs, SCOs or other derivative products that they might include single or multiple pieces of proto-content. The proto-content could be embedded within derivative content, which could contain multiple proto-content objects or no proto-content at all.

Updating content requires a different model than creating it because the content has already been disseminated. It may be difficult to anticipate the nature and scope of changes, and developing a model for update will depend heavily on business rules. A generic

model would presuppose a human in the loop functions as the control authority.

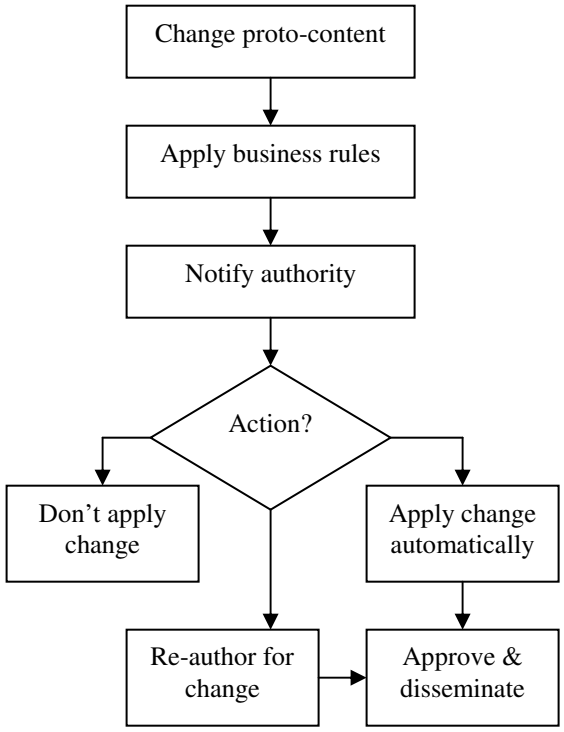

Figure 6. Proto-content change notification

In the general model shown in Figure 6, there are three possible actions in response to a notification of a change to proto-content:

1. Reject the change and keep the derivative content as is (changes are minor; product is due for replacement soon, etc.)
2. Accept the change and let it flow through to derivative content automatically (simple change that has no effect on meaning within context and poses no safety- issues). The change is subject to approval prior to dissemination.
3. Re-author the derivative content (involves a safety issue, change would alter the meaning of the derivative content, etc.) The change is subject to approval prior to dissemination.

We also identified a number of key requirements for proto-content files:

- Format proto-content in a generally readable format that is accessible throughout the organization.
- Use one flexible file format in the proto-content, when possible.
- Choose a proto content file format that can embed other media without a need for links. We used Multipurpose Internet Mail Extension HTML (MHT) files.
- Develop an automatic process to transfer the proto content as a reference to the correct frame in the SCO when proto content changes. As SCOs have mainly unstructured data, it is not an obvious task to automatically update a SCO from an arbitrary proto-content object. If an update should be made to the training part, it can be done manually by the instructional designer.

LESSONS LEARNED

We have compiled the lessons we learned in working through these Use Cases as a set of principles that Program Managers should consider when leading projects that integrate technical documentation and training. Some of these are described briefly below (many more are on the test bed website, www.s1000dscorm.org).

Use Consistent Language Style and Content across Technical Publications and Training

When writing S1000D content that will be used in both technical documentation and training, use language appropriate to both end uses. The typical technical publication constructs won't be applicable to both. Traditionally technical writing is often written in the passive voice. For example, a typical sentence in a technical publication might say: "The bolts must be tightened to 80 foot pounds of torque."

A more contemporary approach would use the active voice and address the user in a second-person, informal style: "Tighten the bolts to 80 foot pounds of torque." While the second sentence is more consistent with the writing style of training, it does not contain the additional information one would hope to find that explains the importance of tightening a bolt to the proper torque, what torque is, how to use a torque wrench, etc. Whereas the technical manual presumes the user is familiar with the concepts and tools and only needs to be told what to do, the training will need to ensure that the underlying fundamental concepts are communicated.

Identify Links to Detailed Content

To use the same content in both places, it will be necessary to identify links to detailed content that provides additional relevant information. This information could be embedded in the training that references the step about tightening the bolts, and links to the training details could also be provided within the technical manual.

The style specifications for S1000D content take into account how the content could be presented in both technical publications and training. When pulling content from S1000D to be inserted into training, the content could be pulled from one of many S1000D forms:

1. as a record in a database within an authoring environment (a BLOB) or XML file,
2. as a record or file that is authorized,
3. as an authorized, published record or file available through a viewer to a user, and
4. directly as a registry of available publications, from whence it can then be found in its CSDB.

We developed a simple viewer that transforms the XML into a representation that can be viewed through an HTML browser. The HTML is not pre-published, but is generated from the XML when the training requests content from the technical publication.

Define Business Workflow before Identifying the Technical Solution

Generally, the critical step of integrating SCORM and S1000D content is to define the business process or workflow that will be used within an organization and then construct the technology to follow the required workflow. We observed this pattern under several scenarios, and once we developed the workflow by which the technical writers or instructional designers specify how to share similar content, the technical solutions can be achieved. There are several key implications:

1. Promote and support early and ongoing collaboration between tech writers and training developers.
2. Anticipate and facilitate cultural and organizational change.
3. Implement a shared infrastructure for lifecycle change notification.
4. Seek a solution that can also be applied to lifecycle logistics data and acquisition strategies.

POLICY GUIDANCE

A number of best practices emerged through development of the use cases. These and other best practices are found in test bed web site, www.s1000dscorm.org; examples are presented here in summary form.

Recommendations for Policy Makers

Effective policy implementation at high levels will enable a smooth transition to a durable approach, whereas a lack of policy leadership will make success elusive. Some example recommendations for policy makers include:

1. Build solutions that leverage open, international standards.
2. Enable organizational determinations of where S1000D and SCORM metadata can be aligned.
3. Require that DM codes have a global unique identifier as required by the Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA).
4. Encourage a facility for inserting and retrieving global variables in and out of LMS.
5. Encourage development of a common CSDB API for access to source content.

These recommendations emerged from close collaboration with experts involved in and capable of influencing the standards

BEST PRACTICES FOR PROJECT MANAGERS

Project managers will play a critical role in achieving the lifecycle cost benefits of integrating S1000D technical publications with SCORM training. We identified a number of key focus areas for facilitating maximum integration.

Get the project off to a good start

To integrate technical publications and training, first specify the workflow or business process that will work within the involved organizations, and then develop the technology to match. The integration process will not be accepted unless it matches organizational needs. When designing the workflow, also consider the cultural shift in the when people who were working separately (technical writers and instructional designers, for example) must now work together.

Implement a conceptual break before and after publishing. In considering business rules and workflow, projects can separate the parts that involve pre-publishing work and post-publishing work. Publishing, the point at which content is made available to others, is a big break. Different sets of business rules and constraints affect these two processes.

Define the financial arrangements. At a project level, new workflow processes must be assigned to some cost unit for payment. In integrated development, the organizations that had previously worked separately but are now working together must understand which organizations will pay for the new processes.

Determine Grain Size. A critical decision within technical publications and within training is the grain size to be used within each domain. The decision balances (a) specificity of content—smaller grain sizes provide more tightly controlled and specified content; and (b) overhead of metadata—large grain sizes allow less use of metadata and minimize overhead.

When developing integrated technical publications and training, the issues are whether the grain size of the technical publications or the training should change and what should be the grain size of units of S1000D embedded in training. Generally, as long as you can clearly specify the content to pull from the technical publications, the grain size of DMs and SCOs should not change when integrating technical publications and training. In developing the method to specify transfer units, the method should work for both the current version of the source, and future revisions.

When considering the content that should be pulled from one source into training, the decision should always carefully consider the question "what degree of surrounding context should be presented with this unit of training so the learner can successfully gain the instructional objective?" This question arises both within the concept of SCORM and SCO reuse, as well as embedding S1000D content in training.

Address the Need to Support Embedding Content from S1000D into Training

There are two practical general-purpose approaches to embedding S1000D into HTML-based training: develop ways to insert content from S1000D into hand-crafted HTML web pages, or develop methods to insert content from S1000D into HTML templates. The hand-crafted solution is best when the instructional design requires flexibility. Hand-crafted training provides more flexibility, but is more time consuming. Templates work best when the training will follow a consistent pattern. Organizations that use templates will frequently want to easily modify the templates they have.

When developing methods that specify the S1000D to be embedded in training, the instructional designer needs a powerful tool to specify what parts of a DM or DMs to include. While the flow of technical publications presents the procedures a user uses when working, the flow of training may not always follow the procedures used when working. For example, when the point of the training is to demonstrate a principle or provide an overview of a procedure, the training will not necessarily follow the steps to complete a task. The training might (a) follow main steps, not stopping to discuss lower level steps; (b) focus on required equipment in various stages of a procedure; or (c) give examples from different contexts for a similar concept. A software tool could be built that supports instructional designers in making flexible references to sections of a DM that could be included in training.

When integrating S1000D content and SCORM training, instructional designers may wish to specify a long sequence of <para>s to be included in instructional presentation by a SCO. As a mechanism and structure for supporting this, we recommend a capability where the instructional designer specifies a start point and a stop point for a set of contiguous <para>s that should be embedded in training. Then, if a new <para>is inserted within the start-stop range, the newly inserted <para> would be automatically embedded in training.

When embedding technical publications into training, the S1000D content to be embedded will often consist of a series of procedural steps (often specified in <para> tags). Associated with these steps is content at the beginning of the DM, the *Warnings, Cautions, and Notes*. The instructional designer should decide how to incorporate changes to them in training. One way is to present the approver with the relevant *Warnings, Cautions, and Notes* that address each <para> included; the approver could then decide and specify if the training should present them.

When S1000D is dynamically linked to and embedded in training, the S1000D content can be made available to training in different formats such as PDF or HTML. Generally the S1000D is available in the XML format, which requires transformation before it can be presented. Using S1000D in ready-to-view format eases real time processing requirements, but using S1000D in XML enables future applications which may require more interactivity. For example, if the S1000D receives values from sensed inputs or dynamic state variables, these values will be easier to insert in dynamically generated presentations rather than in stored presentations.

Another viewer issue (besides input format) is the method for presenting content. A viewer could present the S1000D content in the instructional HTML through use software containers like iframe or with JavaScript. The advantage of using an existing container is ease of development. The advantage of constructing JavaScript is that the S1000D can be presented to stylistically match the style of the surrounding training. The style specifications for S1000D content should take into

account how the content could be presented in both technical publications and training.

To develop effective embedding of S1000D in training, instructional designers and technical writers should have support software. This software should at least:

1. Identify section(s) of the DM to include in training.
2. Preview the training with the embedded S1000D content. This would enable the instructional designers to see what the students would see, making it easier to check the quality of the training.
3. Facilitate showing the S1000D content, from which data could be pulled to fit a need in the training. This helps authors easily locate and create instructional content that could be placed in the training with a presentation style consistent with existing local training.

Enable Assembling of Training Content from Multiple Sources

Address the general need to assemble content for training by selecting specific content from multiple repositories. Pulling content from multiple sources and assembling them into a single new document is a general capability critical to the S1000D SCORM Test Bed. This capability has much wider use. For example, if a web page combines content from a learning vendor company, a simulation company, and a graphics company, content must be pulled from those multiple sources and assembled.

Use Simple Sequencing for Equipment Specificity

Consider using SCORM simple sequencing to support equipment version specificity. Simple sequencing provides a capability for a learner to receive training that is specific to the version of the equipment the learner must learn to operate. This capability combines SCORM Simple Sequencing rules and student profiles. Fundamentally, this provides equipment-specific training using conditional statements in SCORM that have capabilities similar to the S1000D conditional capabilities using *applicability* and *state variables*

In order for SCORM simple sequencing rules to direct training the way *applicability* tags direct IETM presentation in technical publications, real production work should use tools to link training and technical publications. Such a support tool would:

1. Interpret the S1000D to determine the equipment variables that can vary over versions of the equipment and use the variables for simple sequencing rules.
2. Allow instructional designers to see the conditional branching, including factors (e.g., "wheel attachment to frame"), levels of factors (e.g., quick release versus standard release), and the SCO(s) to present for each level.
3. Translate the layout to conditional statements that execute in the SCORM Run Time Environment.

Remember that it goes both ways: linking training to S1000D supports training for IETM users just as linking S1000D to training enables including S1000D content in training. This capability can be useful when a user who is expected to follow an S1000D procedure requires more training to complete the task.

Users of IETMs may need additional training on instructional objects that are not linked to the DM. For example, when following a technical publication for checking a wire in a radio, the DM being used may only link to training about the radio. This link would be created based on the use of that DM in the instructional object. Still, it is possible that the user may need training not on a radio, but on how to interpret a diagnostic reading. Thus, authors may need to add more links between instructional objects and DMs since we want a table that shows all instructional objects that *might be* relevant to understanding the DMs.

Specify the source CSDB

When pulling content from S1000D into training, specify the source CSDB. While the contents of a DM would be expected to be similar regardless of the CSDB where they are stored, they might differ, based on local conditions. For example, different depots may have different tools, leading to different procedures, so the process of pulling content needs to specify more than the DMC, but the appropriate CSDB as well.

To identify the CSDB from which to pull content for a learner, use the learner's profile to identify the appropriate CSDB. Production work should use a tool that helps the instructional designers and technical writers complete the following processes:

1. Specify the DMCs that would be used to supply content to training
2. Specify the relationship between the learner profile, and the important characteristics for pulling the appropriate training for that profile
3. Specify the CSDBs that have the right content for the local training

Implement Tracking/Notification

Develop tracking/notification mechanisms for identifying the relationships (in both directions)

between source content (content that is pulled to be embedded in other documents) and its use in other content. Coordinate this with efforts to implement registry metadata and conform to the trust model.

Implement a robust mechanism for maintaining the authority of content. The method by which integrated content is authorized, revised, and approved will be different with an integrated process than it was with separate technical publications and training development. When S1000D is embedded in training, the primary approval process that must be determined at a project workflow level is "how will changes to S1000D content be reviewed for their effects on training?" A secondary change in the approval process would address the question, "Since S1000D content will be used in training, should it (or at least the technical publications that are used in training) be reviewed by instructional developers?"

The instructional designer should authorize the presentation after development and revisions of content, and authorize its use for training. We recommend following a process similar to what is used in S1000D: Within S1000D, version management is contained within the IDSTATUS section of each DM in the issue number <issno>, issue date <issdate>, and issue type <isstype> fields. In addition, the <issno> element has an "inwork" attribute used for tracking an issue of a DM for drafting. Typically an S1000D authoring environment will modify the *inwork* code for a DM as it goes through the approval routing specified for the project until release. Business rules state the workflow by which notifications and approvals are required to authorize changes in the S1000D. Frequently, authoring environments involve a suite of tools which comprise a content management system that supports version control and approvals. For SCORM, each project would have workflow processes and business rules that specify how a product is ready for release. Similarly, revisions to the training or the underlying S1000D content should trigger requirements for approval, as it does for S1000D and technical publications.

When revising trusted source, some systems use an approach which specifies the complexity of a change. This can be used in the authorization cycle: if a change is minor (e.g., fixing a misspelling), the revision may note that this is a trivial change. An organization may choose how such a trivial change is reviewed and the new version authorized. Other changes are marked as more extensive, and require a fuller review for authorization.

CONCLUSION

The S1000D SCORM Test Bed project demonstrated various ways to integrate S1000D-based technical publications with SCORM-conformant training. Technical efforts to achieve effective integration for large-scale projects will require companion policy level efforts at the organization's highest levels. Program managers must leverage S1000D and SCORM as international, open standards for managing and integrating lifecycle logistics data in order to reduce the lifecycle cost of data, enable interoperability, and further promote data readiness.

REFERENCES

The current version of the S1000D International Specification for Technical Publications Utilising a Common Source Database can be downloaded from http://s1000d.org/.

The current version of the Shareable Content Object Reference Model (SCORM) can be downloaded from http://www.adlnet.gov/.