ADVANCED DISTRIBUTED LEARNING

ADL GUIDELINES FOR
CREATING REUSABLE CONTENT WITH SCORM 2004

JULY 31, 2008
VERSION 1.0 FOR PUBLIC COMMENT

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

Using this Guide

You will learn about the Guide structure, objectives, and intended audience in this section.

Objectives

This Guide supplements the SCORM 2004 documentation suite; *MIL-PRF-29612B Performance Specification, Training Data Products*; *Department of Defense Instruction (DoDI) 1322.20 Development and Management of Interactive Courseware (ICW) for Military Training;* and 1322.26 *Development, Management, and Delivery of Distributed Learnin*g. It provides guidance for instructional designers in implementing SCORM 2004 and registering content in the ADL Registry. You can apply many of the concepts in this Guide to other versions of SCORM content as well, with the exception of *Section 7–Sequencing Your SCORM Content*, which only applies to SCORM 2004.

Intended Audience

This Guide is intended for training development project team members, especially instructional designers, who design SCORM-conformant e-learning and/or develop and register metadata for content in the ADL Registry. This Guide assumes a working knowledge of the Instructional Systems Design (ISD) processes and methodologies.

Conform, Comply, and Certify

Since SCORM 2004 was published, several specifications in SCORM documentation have been accepted as IEEE standards. When using standards, the appropriate term is compliance. You either comply with the standard, or not. The path to certification is compliance. Conformance is not an option. Accordingly, ADL will be amending and updating publications to reflect SCORM-compliance. The SCORM Conformance Test Suite will be renamed as the SCORM Test Suite. This change will be reflected in a future reissuance of DoDI 1322.26. ADL recognizes that there are many documents within our archives that cite conformance. Most are in .pdf or other file formats that cannot be changed. The *ADL Guidelines for Creating Reusable Content with SCORM 2004* is the first instance of this change in terminology.

Commenting on this Guide

This Guide is intended to be a living document. New versions will be released as SCORM evolves and processes are updated. The ADL Initiative also wants to ensure the documents meets the needs of its target audience. If you have comments about the content contained in the Guide or suggestions for improvement, please send an email to ADLGuideComments@adlnet.gov. Include your contact information so an ADL staff member can reach you for clarification.

Summary of Changes

This section will include a list of the changes that were made each time a new version of this Guide is released.

Guide Structure

Figure i.1 depicts how this Guide is comprised of 13 sections that are organized into four high-level groupings: introductory content, background SCORM content, SCORM implementation content, and related content.
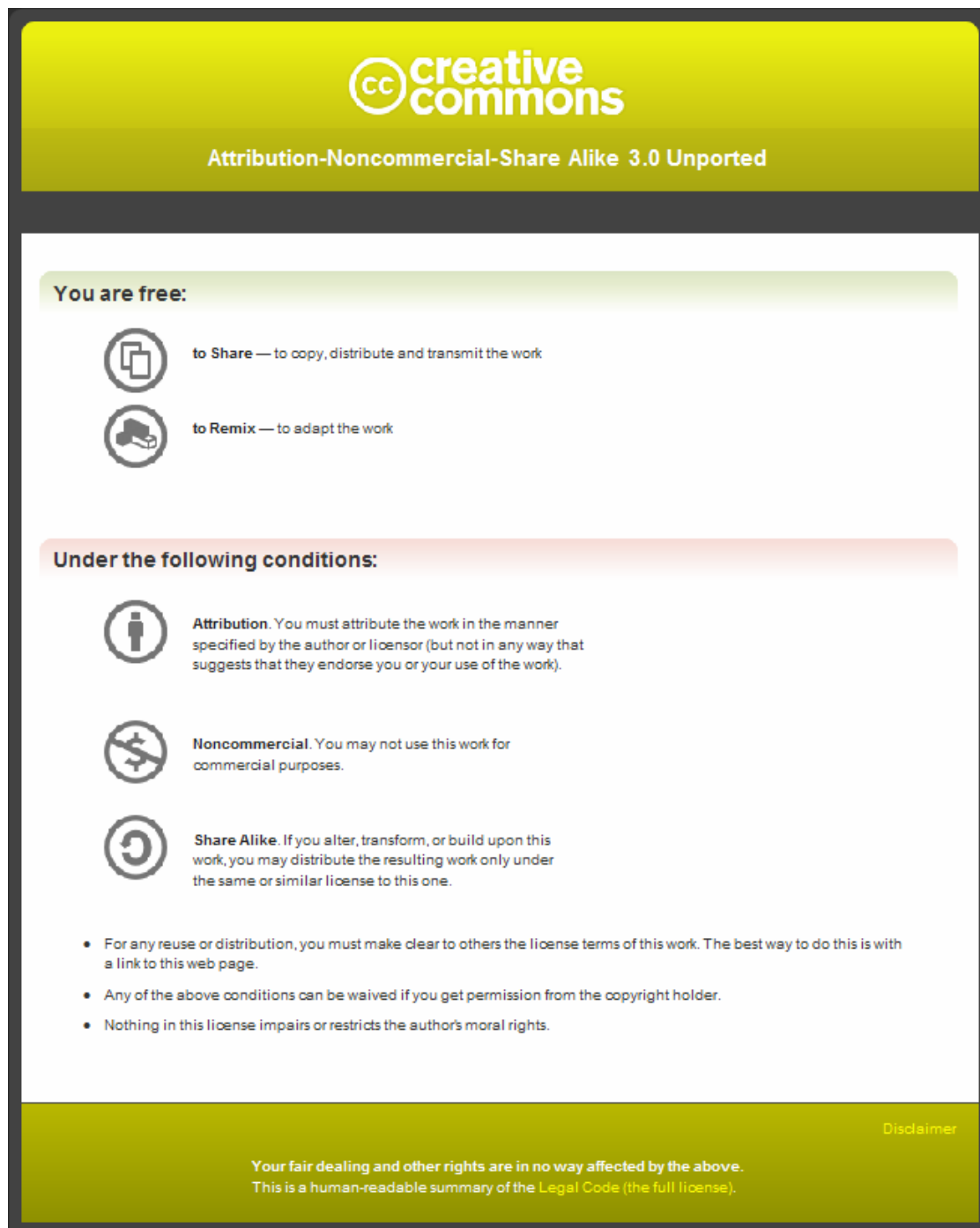
Figure i.1 ADL Guide Structure

1. How do I use the Guide and what's in it? What is ADL and SCORM?

Preface
Using this Guide

Learn about the
Guide structure,
objectives, and
intended audience.

Section 1
**Introduction to ADL
and SCORM**

Learn about the ADL
vision and get a
SCORM overview.

2. What do I need to know to get started with SCORM?

Section 2
**Designing Reusable
Content**

Learn about
creating reusable
content by
redeploying,
rearranging,
repurposing, or
rewriting to meet the
needs of your target
audience.

Section 3
**Defining SCORM
Components**

Learn about LMS
requirements and the
role of assets, SCOs,
aggregations, and
organizations and how
they relate to one
another in the context
of SCORM.

Section 4
**Incorporating SCORM
into Your Process**

Learn about the
impact SCORM may
have on your project
team and design and
development
processes.

Section 5
**Understanding SCORM
Data Model Elements**

Learn about
collecting and
storing data about
learner performance
in, and interaction
with, instructional
content.

3. How do I implement specific SCORM capabilities?

Section 6
**Using Assessments in
SCORM Content**

Learn about
structuring SCORM
assessments as single
or multiple SCOs and
get an overview of
test banks and
cmi.interactions.

Section 7
**Sequencing Your
SCORM Content**

Learn about
structuring your
content for
sequencing,
including the origins of
sequencing,
variables, rules,
modes, and
conditions.

Section 8
**Packaging Your
SCORM Content**

Learn about the
content package
sizing and assembly,
and the manifest file.

Section 9
**Testing Your SCORM
Content**

Learn about the
difference between
conformance and
certification and get
an overview of the
test suite and SRTE.

Section 10
**Creating and
Registering Content
Metadata**

Learn about
searching the ADL
Registry and writing
good metadata to
contribute objects to
the Registry.

4. What are some detailed examples, definitions of terms, and sources used?

Section 11
Glossary

Learn about
definitions for
content described in
this Guide.

Section 12
References

Learn about
reference resources
that contributed to the
content in this Guide.

Section 13
**Authors and
Contributors**

Learn about the
authors of and
contributors to this
Guide.

Licensing and Redistribution Guidelines

Except where otherwise noted, this work is licensed under the Creative Commons License shown below. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to:

Creative Commons
559 Nathan Abbott Way
Stanford, California 94305

This page intentionally left blank.

1   Introduction to ADL and SCORM

You will learn about the ADL vision and architectural functionality, and get a SCORM overview in this section. ADL strives to improve education and training via learning technology. SCORM enables web-based learning systems to find, import, share, reuse, and export learning content. Creating SCORM-compliant content allows you to redeploy, rearrange, repurpose, and rewrite your content to meet your learners' needs.

1.1   ADL Overview

The ADL Initiative was established in 1997 to standardize and modernize training and education management and delivery. The Department of Defense (DoD) Office of the Under Secretary of Defense for Personnel and Readiness (OUSD P&R) oversees the ADL Initiative. The vision of the ADL Initiative is to provide access to the highest-quality learning and performance aiding that can be tailored to individual needs and delivered cost-effectively, at the right time and in the right place.

The ADL Initiative developed SCORM and the ADL Registry. ADL uses structured and collaborative methods to convene multi-national groups from industry, academia, and government who develop the learning standards, tools, and content.

SCORM integrates a set of related technical standards, specifications, and guidelines designed to meet SCORM's high-level requirements—accessible, reusable, interoperable, and durable content and systems. SCORM content can be delivered to your learners via any SCORM-compliant Learning Management System (LMS) using the same version of SCORM. Table 1.1 provides a brief explanation of and example for each of the four architectural functionality requirements addressed by ADL (also referred to as the "ilities").

> For additional information regarding the ADL Initiative, SCORM, and the ADL Registry, refer to the following web sites:
>
> - ADL Initiative and SCORM: http://www.adlnet.gov
> - ADL Registry: http://adlregistry.adlnet.gov

Table 1.1 SCORM High-level Requirements

Requirement	Explanation	Example
Accessibility	Content can be located and accessed from multiple locations and delivered to other locations.	A content author can search the ADL Registry and identify relevant content that has already been developed by another organization and deploy that content on an LMS to learners anywhere in the world.
Interoperability	Content operates across a wide variety of hardware, software, operating systems, and web browsers regardless of the tools used to create it and the platform on which it was initially delivered.	Content packaged for delivery in one SCORM-compliant LMS could be loaded into another SCORM-compliant LMS for delivery to learners.
Durability	Content does not require modification to operate as versions of software systems and platforms are changed or upgraded.	Upgrading to a new computer operating system should have no impact on the delivery of content to learners.
Reusability	Content is independent of learning context and is able to "stand-alone". It can be used in numerous training situations or for many different learners.	E-learning content designed for one organization can be redeployed, rearranged, repurposed, or rewritten by other organizations that have similar learning needs.

1.2   SCORM Documentation, Test Suite, and Sample Run-Time Environment

Table 1.2 provides a brief explanation of each of the SCORM documents, the Sample Run-Time Environment (SRTE), and the SCORM Test Suite (TS). Refer to the SCORM documents for the technical functionality that systems must implement to achieve compliance.

Table 1.2 SCORM Documents, SRTE, and Test Suite Summary

SCORM 2004 Document/System	Explanation
SCORM 2004 Overview	Read this document for an overview of SCORM 2004—the history, current status and future direction of ADL and SCORM, and an introduction to ADL's architectural functionality. URL: http://adlnet.gov/downloads/DownloadPage.aspx?ID=237
SCORM 2004 Content Aggregation Model (CAM)	Read this document for details on how to package content to enable exchange between and among systems, how to describe content for search and discovery, and how to encode sequencing in your manifest. URL: http://adlnet.gov/downloads/DownloadPage.aspx?ID=237
SCORM 2004 Run-Time Environment (RTE)	Read this document for more information about the SCORM Data Model Elements that enable the collection and storage of data about learners' performance in, and interaction with, instructional content. URL: http://adlnet.gov/downloads/DownloadPage.aspx?ID=237
SCORM 2004 Sequencing and Navigation (SN)	Read this document to understand how sequencing can be applied to content to prescribe the manner in which learners receive content from the LMS interoperably. URL: http://adlnet.gov/downloads/DownloadPage.aspx?ID=237
SCORM 2004 Sample Run-Time Environment (SRTE)	Use this prototype LMS to view your content as the learners would see it. URL: http://adlnet.gov/downloads/downloadPage.aspx?id=280
SCORM 2004 Test Suite (TS)	Use the TS to verify SCORM 2004 compliance for your LMS, content packages, and SCOs, and obtain a detailed log of each test outcome. URL: http://adlnet.gov/downloads/downloadPage.aspx?id=277

1.3   Changes from SCORM 1.2 to SCORM 2004

You may hear references to SCORM 1.2, an early version of SCORM. SCORM 2004, the current version, introduces a new set of capabilities that were not available with SCORM 1.2. Table 1.3 provides a brief overview of the changes to SCORM between versions 1.2 and 2004. Not all organizations have upgraded from SCORM 1.2 to SCORM 2004. Before beginning any SCORM content development project, ensure the target LMS is SCORM 2004 compliant so that you can apply the concepts addressed in this document.

Table 1.3 SCORM 2004 Changes from SCORM 1.2

Topic	Description
Sequencing	SCORM 2004-compliant LMSs allow designers to add rules within the manifest to specify the order SCOs are delivered to learners. You can also apply prerequisites to SCOs to ensure that learners know certain information or can complete a particular SCO before moving to another subject or SCO. Rules can be added to provide remediation based on learners' performance within a particular SCO. Sequencing is addressed in detail in *Section 7–Sequencing Your SCORM Content.*
Data Model Elements	There were multiple levels of SCORM-compliance in SCORM 1.2. LMSs could implement only the required SCORM data model elements, a subset of the defined SCORM data model elements, or all of the SCORM data model elements. This contributed to interoperability problems because one organization may have included SCORM data model elements in their content that were not available in a particular LMS on which they tried to deploy their content. SCORM 2004 standardized the data model elements and requires all LMSs to implement them to achieve SCORM 2004 compliance. SCORM data model elements are addressed in detail in *Section 5–Understanding SCORM Data Model Elements.*

This page intentionally left blank.

2   Designing Reusable Content

You will learn about creating reusable content by redeploying, rearranging, repurposing, or rewriting to meet the needs of your target audience, in this section. The ability to reuse content can result in significant savings because much of the time spent creating new content occurs during research and authoring. After you identify a learning need, you can search an online content registry or repository to discover existing relevant instructional materials. If you discover existing materials, and if proper licenses and permissions can be obtained, then you can reuse that content to meet your own specific training needs.

2.1   Defining Reusability

For as long as instructional designers and learning theorists have been talking about learning objects and reuse, they have also been debating the relevance, applicability, and utility of learning objects and reuse. When people hear the term reuse, they immediately equate it with the direct, un-changed use of what is typically highly contextualized content designed for a specific group of individual learners. Given the cost of designing and developing new content from scratch, any savings that can be generated through reuse has merit. In addition to cost savings from reuse, designing reusable content typically implies creating smaller pieces. Since smaller pieces of content can be assembled in many different ways, they can also enable more individualized instruction than is possible with larger pieces of content.

In the following subsections, several different concepts of reusability are described with corresponding examples of each. These concepts describe how instructional designers combine, present, and alter reusable content, independent of the instructional decisions made about the appropriate learning context and/or audience. You can use Figure 2.1 as a comparison of the different concepts of reusability using a consistent example about changing a flat tire; each concept is also described individually.

Figure 2.1 Comparison of Types of Reuse
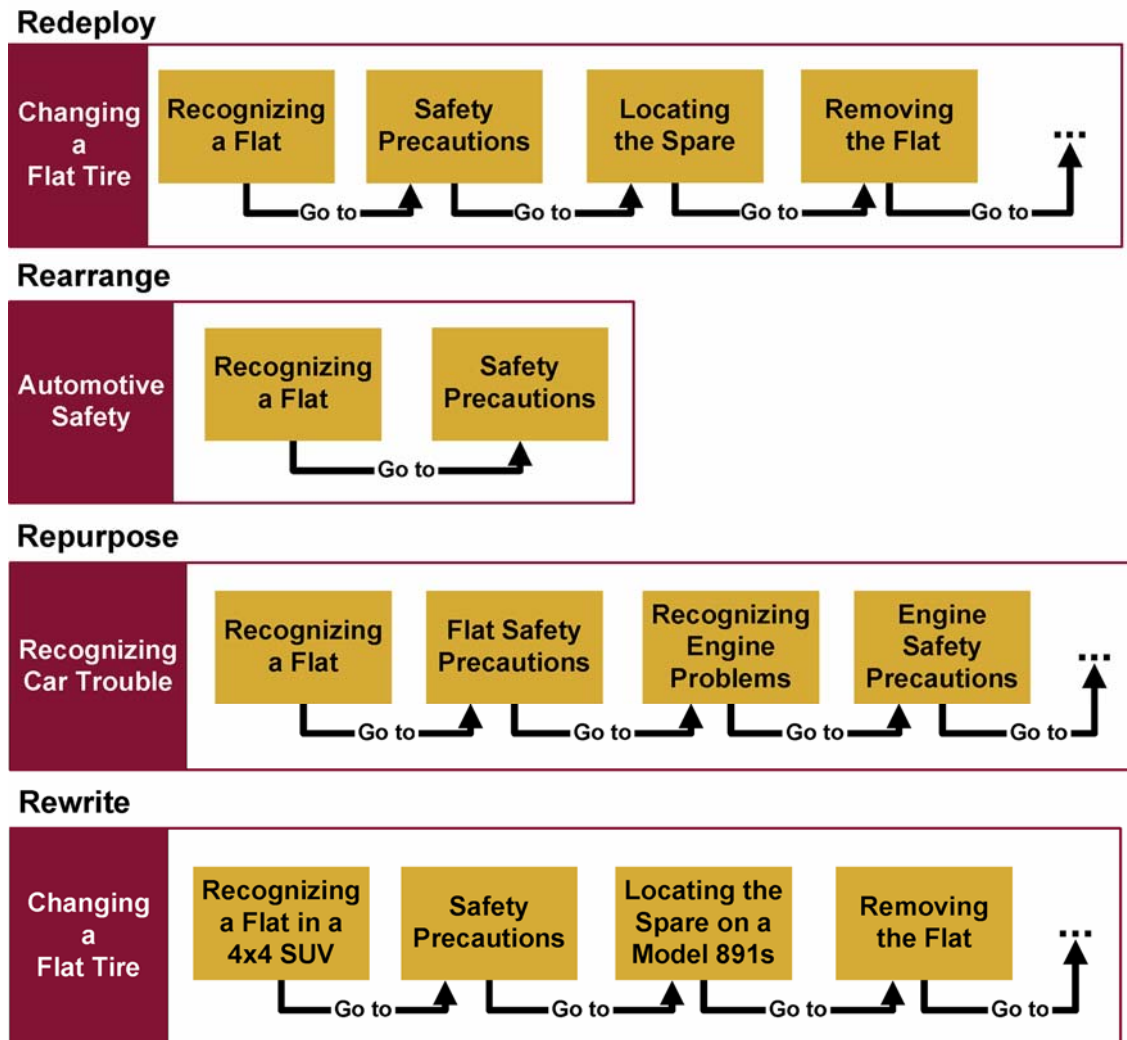
2.1.1 Redeploying to Reuse Content

Redeploying content is running the same content, with no modifications, in multiple Learning Management Systems (LMSs). When content is interoperable, you can use it in any SCORM 2004-compliant system, as it was designed and developed without modification. The Redeploy row from Figure 2.1 depicts generic content about changing a flat tire that could be redeployed in multiple contexts or systems.

For example, a piece of content from an auto mechanics class that teaches learners how to change a flat tire on a car, could be deployed on multiple LMSs in various school districts across the country without modification.

Likewise, a piece of content developed by an automobile association to teach their members about safety considerations when they have a flat tire could be used, unchanged, as part of the driver's education curriculum in a high school, or in a program sponsored by insurance agencies to help drivers lower their car insurance rates.

Redeploy

Changing a Flat Tire	Recognizing a Flat	Safety Precautions	Locating the Spare	Removing the Flat	...

2.1.2 Rearranging to Reuse Content

Rearranging content is taking existing content and re-ordering it for new uses or contexts. For example, the Rearrange row from Figure 2.1 depicts a piece of content developed by an automobile association for Automotive Safety. This content is used to teach members about safety considerations when they have a flat tire. That same content could also be used as part of the driver's education curriculum in a high school, or in a program sponsored by insurance agencies to help drivers lower their car insurance rates.

Rearrange

Automotive Safety	Recognizing a Flat	Safety Precautions
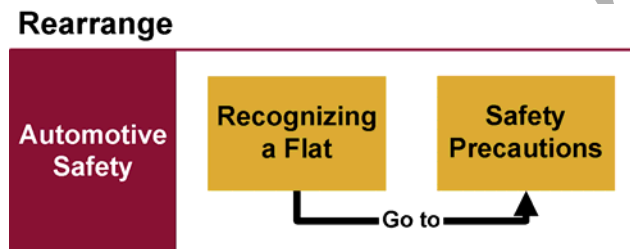
2.1.3 Repurposing to Reuse Content

Repurposing content enables you to take the same piece of content and use it in new contexts or in different ways. For example, the Repurpose row from Figure 2.1 depicts how content about recognizing a flat tire and flat tire safety precautions could be combined with content about recognizing engine problems and engine safety precautions to create a new collection called Recognizing Car Trouble.

Repurpose

Recognizing Car Trouble	Recognizing a Flat	Flat Safety Precautions	Recognizing Engine Problems	Engine Safety Precautions	...
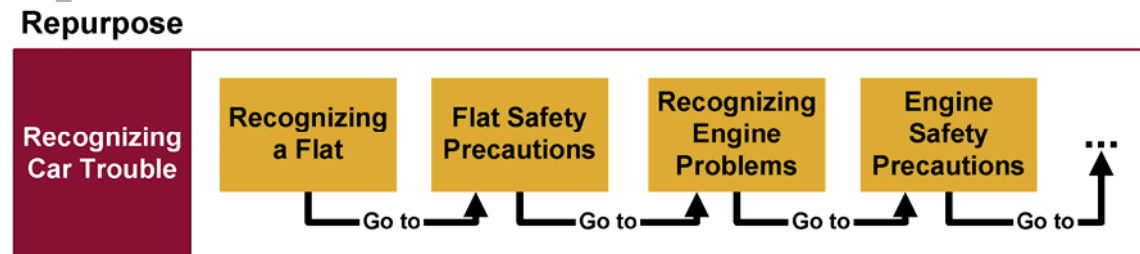
2.1.4   Rewriting to Reuse Content

A significant portion of the time you spend designing new content is devoted to researching and authoring the content. If the content is designed so that it is highly granular, then taking the relevant materials and changing the examples or imagery, rewriting the verb tenses or modifying the person or voice of the subject for different audiences, or removing irrelevant information can save you effort that you can then apply to creating a more valuable learning experience.

The Rewrite row from Figure 2.1 depicts how the content about recognizing a flat tire can be rewritten to specifically target recognizing a flat tire in a four-wheel drive Sports Utility Vehicle (SUV). In this example, the safety precautions content is unchanged, the locating the spare tire content has been rewritten to specifically target car model 891s, and the removing the flat tire content also remains unchanged.
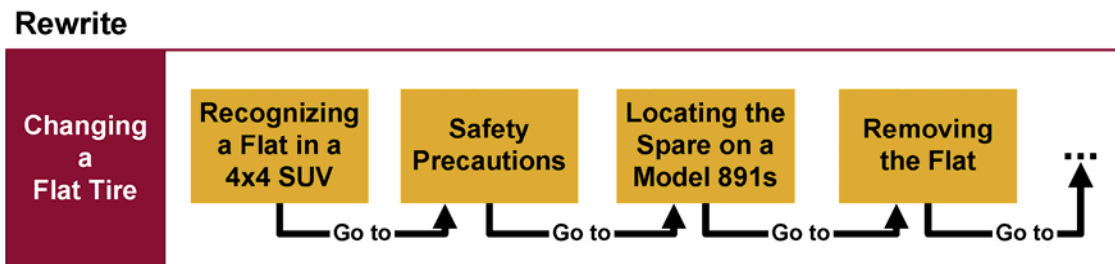
2.2   Understanding the Role of Context

Context is an essential component of learning—people learn best when instruction is presented in the context in which it will be used. Good instructional designs build knowledge incrementally using previous knowledge as the foundation for moving from the known to the unknown. This is typically accomplished with a reference to a previous instruction, such as, "In the last lesson you learned…" Statements such as these actually hamper reuse because they appear within the content and must be removed for it to be reused.

One assumption of reuse is that to obtain the maximum potential to reuse content, the content must be context neutral, or not specific to any one audience, item, or subject area. Since this philosophically contradicts what you may know about designing good content, you may be concerned about the validity of context-neutral learning content.

> **Key Point**
>
> To make a content object completely reusable, avoid explicit references to other content.

Not all content can or should be context-neutral, just like not all content can or should be reusable. However, changing how you think about your content will help you create instructionally-sound, context-neutral, reusable content, when possible, and allow you to weave context into your content by placing the context around the reusable pieces rather than inside them.

Figure 2.2 depicts a portion of content about changing a flat tire on a vehicle produced by USACar Company targeted primarily at their sales force. USACar decided to provide some generic content about how to recognize a flat tire and safety precautions when you have a flat tire, so the sales force can discuss these topics with their customers. These are applicable to any vehicle and also to any particular audience (such as insurance companies, driver's education programs, etc.). However, since USACar produces many different car models with the spare tire located in the trunk, the under-carriage, and the exterior of the vehicle, USACar designed a piece of context-specific content for each vehicle they produce to show salespeople and car owners where the spare tire is located in their specific vehicle.

Figure 2.2 Example of Non-Contextual Content Surrounding Contextual Content
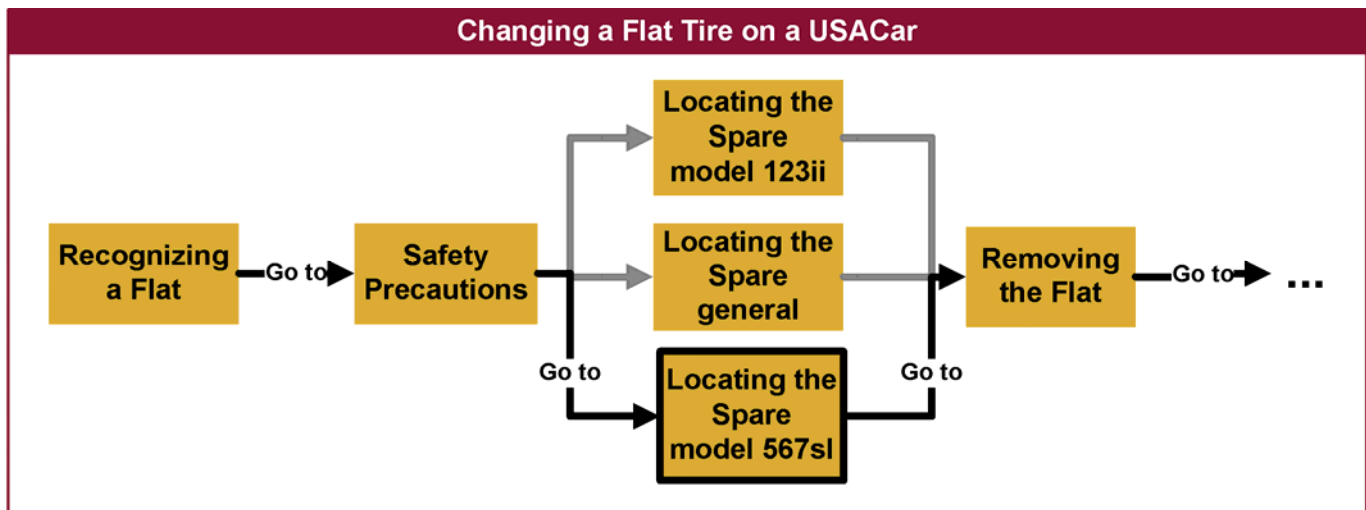
In this example, after learners view the safety precautions content, they are routed to the content for the specific car model that they sell or own. Following the context-specific piece of content about the location of the spare tire in their particular model, they return to the general instruction on how to remove the flat tire. This is an example of how context-neutral content (recognizing a flat, safety precautions, and removing the flat), can be interspersed with context-specific content (the location of the spare tire in a particular car model) to create an instructionally sound learning experience that leverages reusability.

2.3　Designing Reusable Content

2.3.1　Determining Potential Audiences

The first step in designing reusable content is determining which, if any, portions of your content could be useful for someone other than your target audience.

If your target audience for content about flat tires is members of an automobile association, then your content could be designed for all drivers, since members of an automobile association also represent the general public; there are no unique characteristics about them or their driving habits. The same content may be reused in different industries by different learners to achieve the same or even different outcomes.

Likewise, if your target audience is truck drivers who must be certified to transport hazardous materials, you could leverage existing content on types of hazardous material and hazardous materials classification developed or used by emergency responders. In this case, reusing content would be especially beneficial because hazardous materials classification is designated by the United Nations—standardizing it internationally—and Title 49 of the U.S. Code mandates the requirements for hazardous materials transportation in the United States. Since the rules and requirements for this portion of the content are not unique to one trucking company, you may find that the content you are developing can be used by more than just the individuals for whom it was originally intended, or that relevant content that could be useful to your audience already exists. Figure 2.3 depicts (in the highlighted box) some Transportation of Hazardous Materials content that could be reused by both a trucking company and emergency response personnel.

Figure 2.3 Example of Content that can be Reused from Another Source
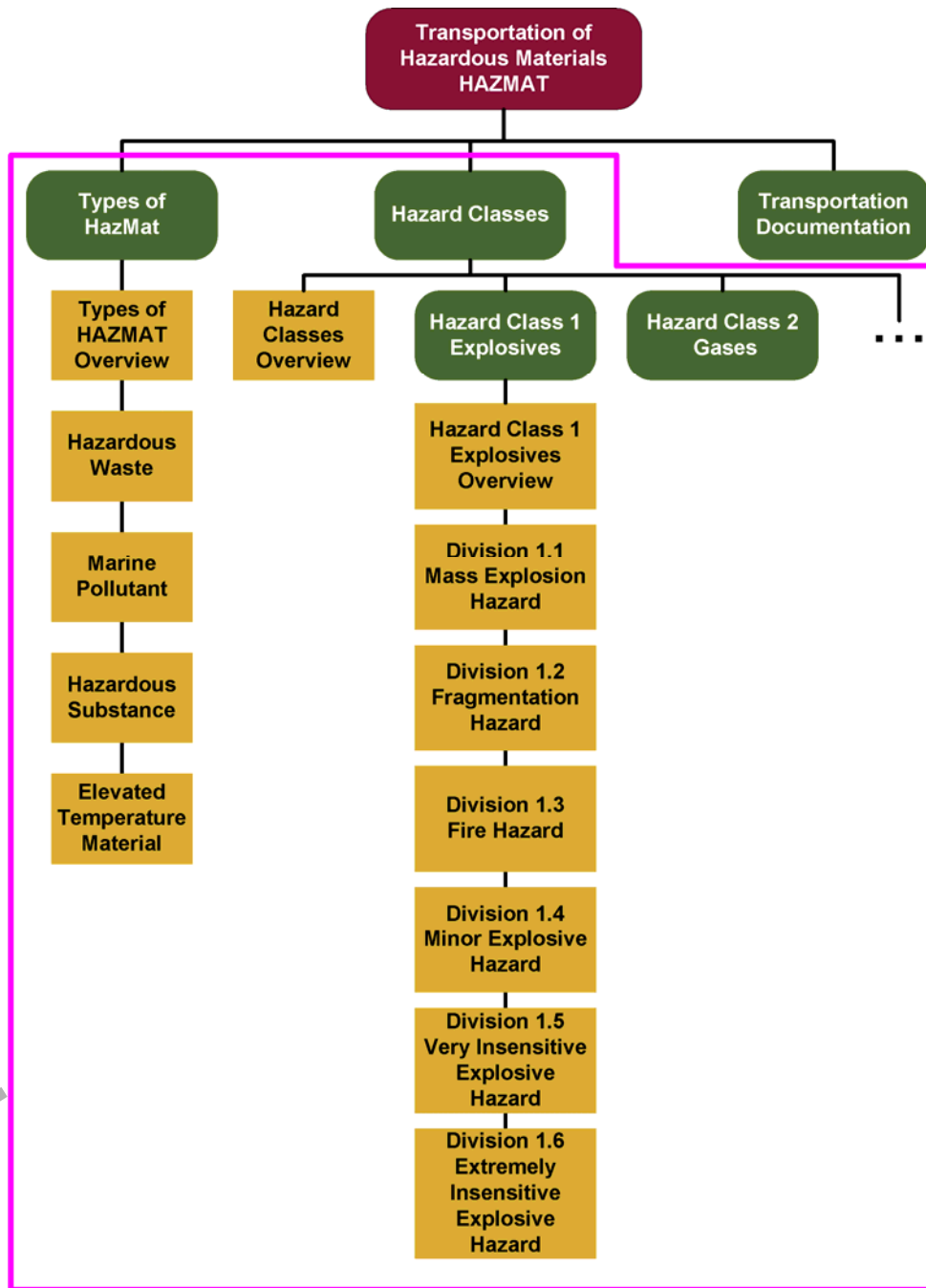
2.3.2   Identifying Reusable Pieces

Since reusable content is intended to be inherently small, a piece of content that addresses a broad subject like Operating Your Car might be too comprehensive to be reusable for most audiences. However, a smaller piece of content, such as, Before Starting Your Engine may be reusable because the procedures you follow are fairly consistent and unrelated to the type of vehicle you are operating.

⊕ **Consideration**

To determine how best to divide or chunk your content, look at the potential audiences you have identified and ask yourself the following questions:

- What portions of this content apply only to my target audience?
- Can the content be divided before and after the portion that applies only to my target audience?

2.3.3   Adding Context

After you've identified the pieces of content that can be reused, you can add context to your content by weaving in pieces that are context specific. When you limit content to a single, granular, well-written learning objective, it is easier to make it context-neutral. Where context-specific instruction is required, you can create context-specific objectives like, "in accordance with the owner's manual, state the location of the spare tire in a USACars Model 891S, without assistance."

2.4   Reusing Content Developed by Others

As a result of your front-end analysis (FEA), you may be asked to reuse content developed by others. The materials you are asked to develop may or may not have been created by your organization. They also may or may not have been developed by instructional designers. When tasked with reusing existing content that is instructionally sound, determine whether you will redeploy, rearrange, repurpose, or rewrite.

This page intentionally left blank.

3   Defining SCORM Components

You will learn about LMS requirements and the role of assets, SCOs, aggregations, and organizations and how they relate to one another in the context of SCORM, in this section.

3.1   SCORM Requirements for LMSs

A Learning Management System (LMS) is a software package that is typically accessible via the Internet and is used to administer one or more courses to one or more learners. Simply stated, an LMS allows learners to authenticate themselves, register for courses, access learning content, take assessments, and then store the learners' performance records.

SCORM does not specify where or how you store and manage your content. You can store SCORM content in any system that stores electronic files. Some examples of content storage tools include content repositories, content management systems (CMSs), LMSs, and Learning Content Management Systems (LCMSs).

SCORM requires that content be accessible via an LMS or an LCMS at "run-time" (that is, when learners interact with the content). SCORM defines certain capabilities that an LMS or LCMS must provide to be SCORM compliant.

While SCORM does not recommend any specific tool, programming language, system, or learning technology architecture, it does define the collection, delivery, and communication methods for content. SCORM does not require systems to look and work alike. However, SCORM specifies how content is created to enable it to be redeployed from one system to another. When deploying SCORM content, the LMS is the primary interface between the learners and the instructional materials. SCORM standardizes the following two principal aspects of an LMS:

- How new content is moved from an authoring environment into an LMS.
- How Sharable Content Objects (SCOs) communicate with an LMS to perform operations such as "get a learner's name" or "set a score."

An LMS may perform or enable optional features such as: authoring, classroom management, competency management, knowledge management, certification or compliance training, testing, personalization, mentoring, video conferencing, chat, and discussion boards. These optional features are not defined by SCORM, so LMS vendors have the flexibility to implement functions in their system in their own unique and proprietary way to maintain the proprietary advantage that differentiates one system from another in the marketplace.

> **Key Point**
>
> If you use functionality not prescribed explicitly by SCORM, you jeopardize content interoperability because you may have difficulty running your content in other LMSs.

For example, SCORM does not address competency management. If you use the competency management function provided by your particular LMS, and then you try to export your content to a different LMS, the competencies you assigned may not function as you intended because SCORM deliberately avoids specifying all of the functionality that a commercial LMS could provide.

3.2   SCORM Content Components

Figure 3.1 depicts SCORM content components, from smallest (assets) to largest (curricula); each component is described individually. The colors you see here for each component are used throughout this Guide to help you quickly identify the types of components. Assets are blue, SCOs are yellow, aggregations are green, and organizations or root aggregations are red. Curricula are shown here in red, but they may be comprised of other learning activities and are outside the scope of SCORM.

Figure 3.1 Components of SCORM Content
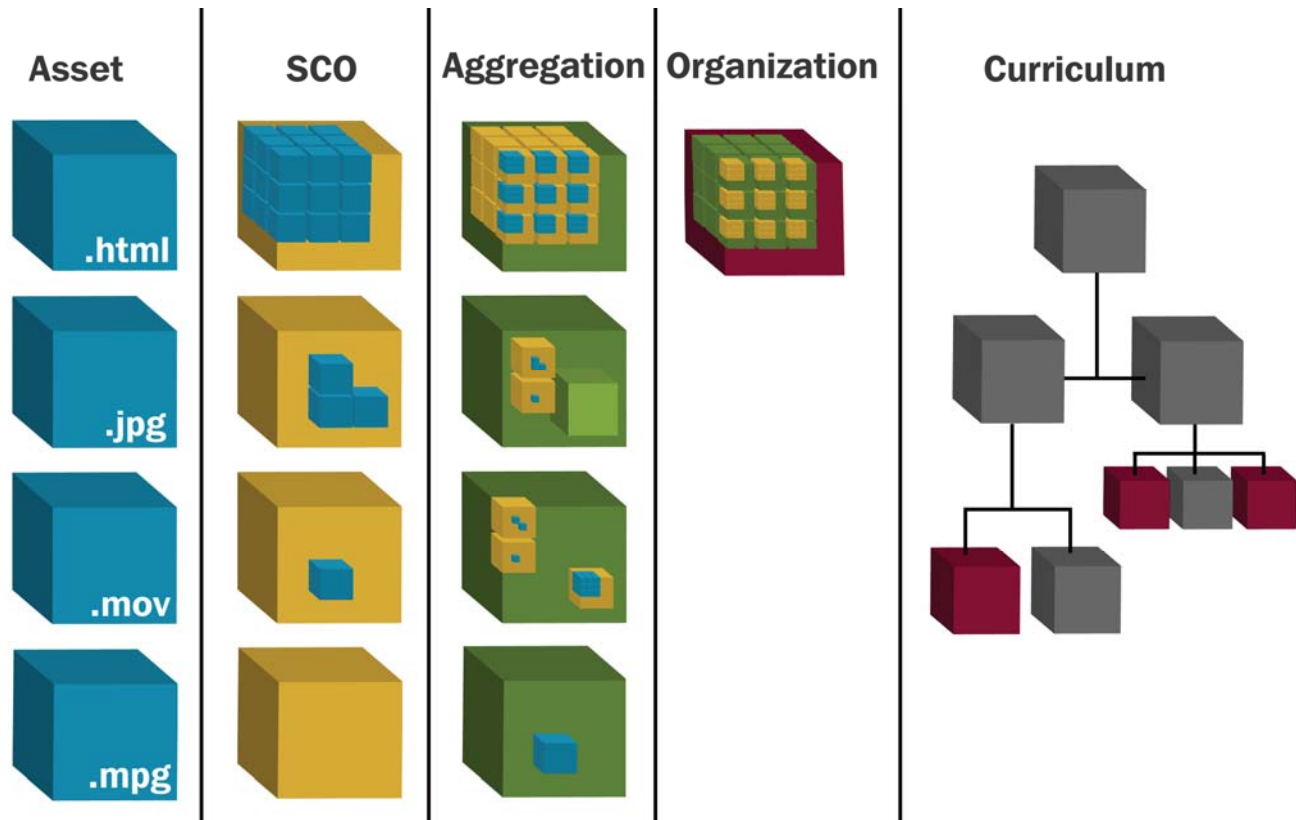
This diagram was derived from a reusability diagram created by Wayne Hodgins of Autodesk®.

3.3   Asset

Assets are electronic representations of media, text, images, sounds, HTML pages, assessment objects, and other pieces of data. They do not communicate with the LMS. Assets will likely be your most reusable items; they can be redeployed, rearranged, repurposed, or reused in many different contexts and applications. The figure to the left depicts each asset as a small blue box with examples of several asset types (such as, .gif, .mpg, .html, .txt, .jpg).

For example, in Figure 3.2, an image of the Hazard Class 7 Radioactive placard could be used in training materials for different audiences in both commercial and DoD transportation as well as by different individuals who may be affected by the transportation of hazardous materials such as truck drivers, first responders, and shipping inspectors.

Figure 3.2 Shared Asset Example

3.4   Sharable Content Object

SCOs are the smallest logical unit of information you can deliver to your learners via an LMS. The term SCO has different implications for instructional designers and programmers. Instructional Systems Designers (ISDs) and content authors view a SCO as content; they focus on the actual instructional material in the SCO. Programmers may view a SCO as a location in a Table of Contents to apply programming rules to, or as a collection of assets that control communications with the LMS.

From an ISD perspective, a SCO is most commonly a defined piece of instruction comprised of one or more assets. Since SCOs collect information on learners' performance, structure your SCOs based upon the specific performance needs of your learners.

If you are providing your learners with broad instruction about handling hazardous materials or another compliance-type subject, your only requirement may be to show that learners have completed the content. In this case, completing the content usually means learners have viewed it, so the content for this course could be a fairly large single SCO.

On the other hand, if you need to know that a truck driver learned to "properly load and store radioactive materials during transit" as part of a larger hazardous materials course, then this objective could be a single SCO that describes this individual aspect of loading and storing radioactive materials in transit. It is likely that this SCO would be one of many smaller SCOs to confirm that the truck driver understood specific aspects of hazardous materials transportation.

In technical terms, a SCO is defined as the only piece of information that uses the SCORM Application Programming Interface (API) for communication with an LMS. The SCORM API is a standardized method for a SCO to communicate with the LMS when learners are interacting with a SCO. There is specific information the SCO can retrieve (GET.VALUE) from the LMS and store (SET.VALUE) in the LMS. For example, it can store values in the LMS, such as a score or completion status, or retrieve information from the LMS, such as a learner's name. Figure 3.3 depicts the API communication link between a SCO and an LMS.

Figure 3.3 API Communication Link
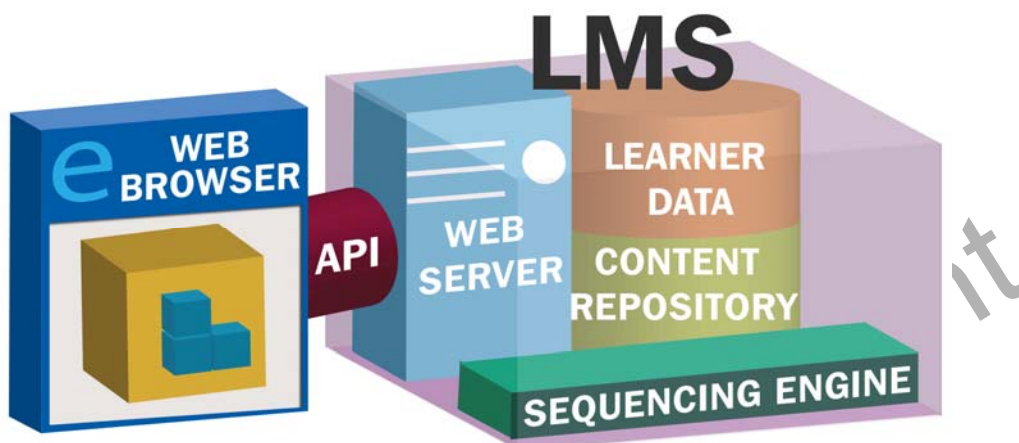
Experienced SCORM designers talk about SCOs as the content that learners see and interact with, but this is not the complete story. Technically, a SCO must communicate with the LMS to be called a SCO, but you can also present the learner with content that does not communicate with the LMS. For example, you can use a simple HTML file or PDF document to present learners with a set of instructions before they interact with the content. Even though this type of non-communicating content does not meet the technical definition of a SCO, it is usually called a SCO when talking about the aggregations or sequencing rules described in subsequent sections.

3.5   Aggregation

An aggregation is a collection of related content. In this Guide, an aggregation is defined as a parent and its children in a tree structure. The SCORM documents refer to an aggregation as a cluster. The figure to the left depicts aggregations as green boxes and SCOs as yellow boxes containing blue assets. Aggregations are used to group related content so that it can be delivered to learners in the manner you prescribe.

An aggregation is not a physical file; it is a virtual home within a SCORM organization where sequencing rules are applied to a collection of related SCOs or aggregations. Sequencing rules allow you to prescribe the behaviors and functionality of the content within an aggregation as well as how one aggregation relates to other SCOs or aggregations within an organization (root aggregation).

SCORM aggregations can contain SCOs as well as other aggregations—so aggregations of aggregations are possible. Figure 3.4 depicts a SCORM aggregation called Hazard Classes that contains three other aggregations: Hazard Class 1, Explosives; Hazard Class 2, Gases, and a third aggregation showing this structure would continue to reveal individual aggregations for all nine hazardous materials classes. These aggregations, the green boxes, do not contain content; they are merely a way of structuring content to apply sequencing rules.

Figure 3.4 Example of Potential SCORM Content Aggregations
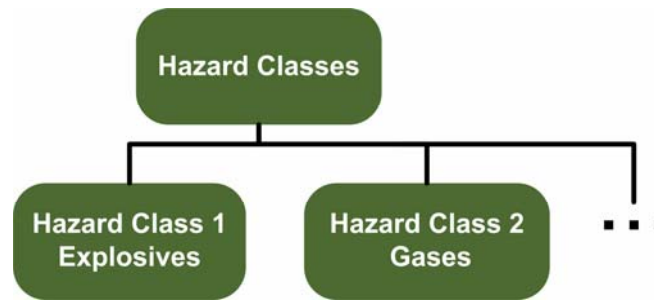
Figure 3.5 depicts the expansion of the Hazard Class 1 Explosives aggregation, the green box, with six SCOs, yellow boxes: Division 1.1 Mass Explosion Hazard, Division 1.2 Fragmentation Hazard, Division 1.3 Fire Hazard, Division 1.4 Minor Explosion Hazard, Division 1.5 Very Insensitive Explosion Hazard, and Division 1.6 Extremely Insensitive Explosion Hazard. Refer to *Section 7–Sequencing your SCORM Content* to learn more about structuring your content for sequencing.

Figure 3.5 Example Expansion of a Potential SCORM Content Aggregation
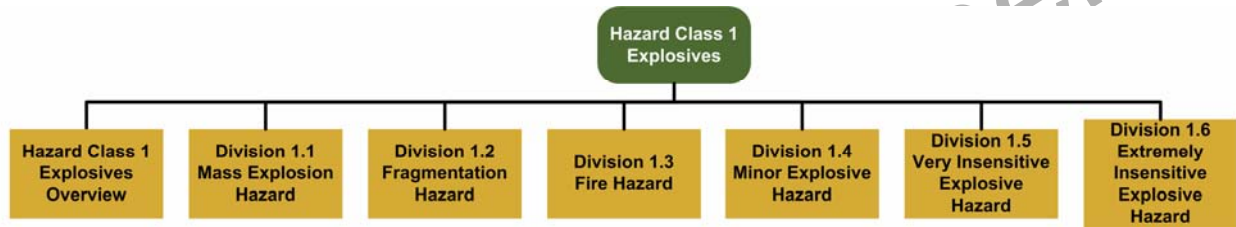
3.6   Organization
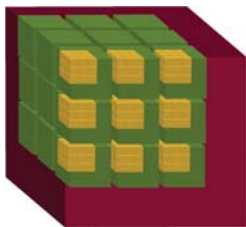
The organization is the part of a content package where SCOs are ordered into a tree structure and sequencing behaviors are assigned to them. The figure to the left depicts an organization as a red box containing multiple green aggregations and yellow SCOS. The organization outlines the entire structure you have created for the content that you intend to deliver as a single content package. Each organization is a top-level aggregation, also referred to as the root aggregation in this document.

Figure 3.6 depicts the organization called Types of Hazardous Materials (HazMat) using a red box at the top of the tree with rounded corners. The organization represents a content package containing three aggregations. The aggregations: Types of HazMat, Hazard Classes, and Transportation Documentation, are green boxes. The Types of HazMat and Hazard Classes aggregations also show their associated SCOs, the yellow boxes, so you can see how organizations are structured.

Figure 3.6 Example of Potential SCORM Organization
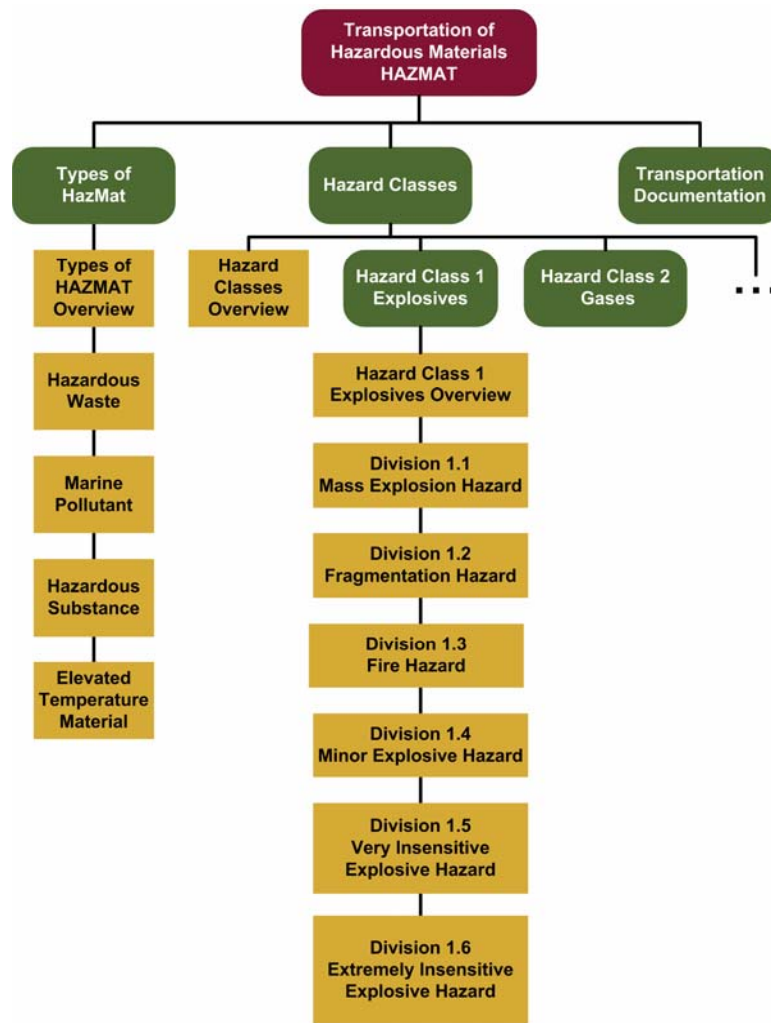
3.7 Curriculum or Course
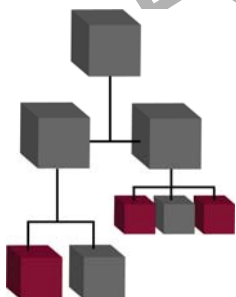
While a curriculum or course is outside the scope of SCORM, SCORM-compliant content can be part of a curriculum or course that is managed by your LMS. The figure to the left depicts a curriculum consisting of multiple independent components. In this figure, the red boxes represent SCORM organizations while the gray boxes represent a combination of other learning experiences (such as, collaboration sessions, labs, lectures). A curriculum typically includes courses, lessons, and assessments using a variety of delivery media and instructional strategies.

The curriculum for a truck driver may include SCORM-compliant content organizations in loading, hazardous materials transportation, and road safety, as well as, actual driving or loading experiences on a test course and also on public roadways. The LMS would deliver and store learners' performance data for the SCORM-compliant content organizations, and it might also store and manage the scheduling and completed work accomplished on the test course and public roadways.

Multiple organizations may appear within a curriculum. Figure 3.7 depicts a curriculum called Truck Driver Certification in the top gray box. The curriculum includes two SCORM organizations: Road Safety and Transporting Hazardous Materials, both shown as red boxes. It also includes a lab activity called Loading Lab, shown in the gray box, that would be performed in a loading simulator or on an actual truck.

Figure 3.7 Example of Potential Curriculum with SCORM Organizations
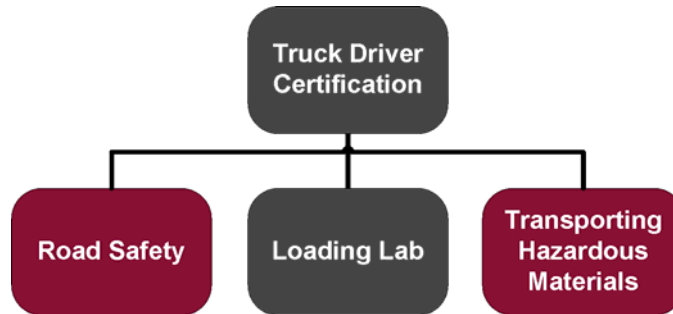
Consideration

When selecting tools, programming languages, and systems, you should conduct a thorough analysis of your requirements and decide on clearly-defined criteria prior to making implementation decisions. Consider how your selection will:

- Integrate with your existing technology infrastructure investments.
- Meet the requirements defined by your customers, internal development team, and learners.
- Provide the functions, interactions, collection, and reporting that you require.
- Satisfy your strategy for the current and long-term content, including:
  - Quantity of content being managed
  - Diversity of contributors to the content as well as learners accessing the content

This page intentionally left blank.

4   Incorporating SCORM into Your Process

You will learn about the impact SCORM may have on your project team and design and development processes in this section. Designing, developing, and delivering SCORM content requires a variety of skill sets and expertise. While your overall process will not change, you may have to account for a few new steps along the way.

4.1   SCORM Team Members

A strong and well-balanced team that adheres to standard design and development conventions is an important success factor, particularly as designs are passed from designers to programmers for content packaging. Clearly defining the roles and expectations of your team is crucial to the success of your SCORM content development projects.

Programmers should focus their efforts on the technical implementation of SCORM. For content to function both technically and instructionally, programmers should advise instructional designers on the technical constraints that govern how SCOs are created and delivered and the application of sequencing rules. The programmers must have an in-depth knowledge of the technology required to implement SCORM and be able to structure data in Extensible Markup Language (XML) to meet the SCORM compliance requirements. Programmers should also create the SCORM content package.

As design-related questions arise, such as, "Can I do this with the content and still make it SCORM compliant?" you will rely on the technical expertise of programmers to explain the technical constraints under which they are working. Likewise, as programmers begin the technical implementation of the instructional materials, they will need to confer with you to ensure that the materials are functioning in a SCORM-compliant LMS as you intended. It is essential that ISDs and programmers form a cohesive team that can work together from the initial planning stages of the project through delivery.

The titles and responsibilities of individuals on your development team may vary from those outlined in Table 4.1, but this provides a general set of roles and responsibilities for a typical team. Likewise, the specific skills each team member needs may depend on the tools you use to develop your content.

> ## Key Point
>
> Instructional designers and content authors do not need to master all of the technical nuances of a SCORM implementation. Their primary responsibility should remain designing effective instructional materials that work within the SCORM technical guidelines. This Guide should provide members of your team with the basic knowledge they need to design and author instructional materials that are SCORM compliant.

Table 4.1 Typical Design and Development Team Member Roles and Responsibilities

Team Member Role	Typical Responsibilities
Program/Project Manager	Educates the client about the benefits of SCORM content delivery. Scopes and defines the level of effort for the content and its SCORM-compliant features. Manages communication with the client. Ensures product is delivered free of defects, on budget. Commits resources to ensure that SCORM compliance is accomplished. Works with ISD to conduct Front-End Analysis (FEA).
Instructional Designer (ISD)	Works with program manager to conduct Front-End Analysis (FEA). Reviews and writes learning objectives. Determines instructional architectures and strategies. Designs basic content structure. Creates design documents, specifications, and initial content outlines. May write some metadata. Tests, validates, and evaluates learning materials. May write assessment items and create assessments. Works with other team members to ensure instructional integrity.
Subject Matter Expert (SME)	Works with ISD to determine appropriate SCO size and instructional strategy. Ensures instructional materials are technically accurate and appropriate for the audience in accordance with client needs and requests. May work directly for the client.
Content Author	Takes basic instructional design and content outline from ISD and researches and writes all instructional text or scripts. May write assessment items and create assessments. May request or design media assets. May write some metadata. May also be referred to as a content developer.
Programmer	Defines sequencing rules and creates content structure specified by ISD in XML format. Modifies existing SCORM sequencing rules, metadata, and manifests in XML, as needed. Creates SCORM content package specified in the design. Works closely with instructional designers to ensure content structure and sequencing behaviors meet SCORM technical implementation guidelines. May also be referred to as a developer.
Quality Control/Assurance (QC/A)	Verifies content complies with organization's procedures, policies, and style guidelines. Conducts first-phase testing to ensure content is error/bug free. May test content in ADL Sample Runtime Environment and Test Suite or target LMS. Maintains ADL SCORM test logs. May conduct formative evaluation.
Content Librarian	Creates, maintains, and approves metadata records for assets, SCOs, aggregations, and content packages. May load assets into content repository or Learning Content Management System (LCMS). May register content in the ADL Registry. Locates materials for reuse during FEA. Defines or maintains organization's metadata taxonomy and controlled vocabularies. Works with other team members to ensure the development and management of accurate, effective, and standardized metadata.
Graphic Artist/ Media Producer	Creates and captures assets (graphics, video, animation, etc.) and user interfaces in accordance with requests from ISDs and content authors. May record and produce sound and narration.
System (LMS) Administrator	Creates and manages user accounts in LMS. Ensures LMS functions properly before delivering instructional materials to learners. May load content packages into LMS. Responds to learner registration issues and problems.

4.2   Design and Development Processes

Building compelling, reusable SCORM-compliant content requires significant coordination, planning, and teamwork. Equally important is establishing clear expectations, processes, and design and development guidelines to ensure the entire team knows what should be done and how it should be done. There are several ISD approaches, but this Guide uses the Analysis, Design, Development, Implementation, Evaluation process (ADDIE). Figure 4.1 depicts which team members typically participate in each phase of the ADDIE process.

Figure 4.1 Team Members Typically Involved in Each Phase of the ADDIE Process
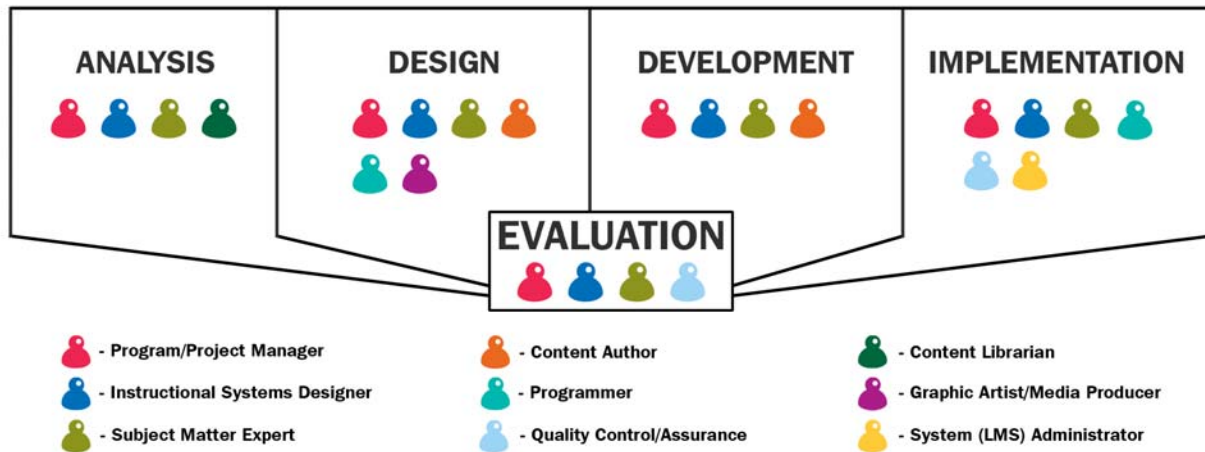
All members of the project team should be aware of the inputs and outputs of each phase. The subsequent sections will help you make sense of how your team might work together to develop high-quality SCORM content throughout all phases of the instructional design process. It also suggests specific SCORM considerations for creating your content.

4.2.1   Analysis Phase

The analysis phase focuses on identifying instructional requirements, desired outcomes, and existing resources. When you analyze the problem that created the need for an instructional intervention, remember to examine the audience's learning needs and the needs of the project, including defining requirements, confirming funding for the project, and drafting a project schedule. Obtain input from all key stakeholders, but designate one as the ultimate authority should it become difficult to reach a decision.

If you are developing content on behalf of the DoD, you must adhere to the guidance provided in *DoD Instruction (DoDI) 1322.26 Development, Management, and Delivery of Distributed Learning. DoDI 1322.26* requires content authors, as part of a FEA, to search for existing content in the ADL Registry before creating new content. The goal of *DoDI 1322.26* is to make quality learning content more widely accessible, and to reduce costs by allowing the redeploying, rearranging, repurposing, and rewriting of content instead of constantly developing new content. You should also search the Defense Automated Visual Information System/Defense Instructional Technology Information System (DAVIS/DITIS) to identify any existing media elements or content components. The DAVIS/DITIS site contains the searchable listings and descriptions of thousands of Audiovisual (AV) productions and Interactive Multimedia Instruction (IMI) products used by the DoD.

> As part of your FEA, refer to the following web sites:
> - DAVIS/DITIS: http://defenseimagery.mil/index.html
> - ADL Registry: http://adlregistry.adlnet.gov/

For your FEA, you will need to document your search results from the ADL Registry and DAVIS/DITIS by identifying what, if any, content can be redeployed, rearranged, repurposed, or rewritten. If you are unable to locate existing relevant content in the ADL Registry, you must document the need to create new content.

If you plan to use content identified in the ADL Registry, your FEA report should state how you will gain access to the content and how it will be incorporated into your instruction. Remember, obtaining permission to reuse content, as well as obtaining the physical files, may be a lengthy process. To avoid delays, you should start early. Do not wait until the FEA is finalized before seeking approval.

⊕ **Consideration**

In addition to the typical activities associated with the instructional design process, consider asking these questions as you analyze the content:

- Have you searched the ADL Registry to identify existing content that can be redeployed, rearranged, repurposed, or rewritten?
- Does the target LMS comply with the latest version of SCORM?
- What data do you need to collect about learner performance in and interaction with the instruction?
- Can you immediately identify secondary audiences for your content to increase its reusability?

4.2.2   *Design Phase*

The design phase is driven by the outputs of the analysis phase. In this phase, designers write learning objectives, plan assessment instruments, devise learning activities, structure content, and request media. If not already documented, you should capture your standards and procedures in a style guide. As part of the design process, the graphic artist should create a Graphical User Interface (GUI) template for layout and navigation to establish the look and feel for the project and you should create a content structure diagram to depict the relationships among content.

Your complete diagram (also referred to as a course map) organizes the SCOs and aggregations and depicts their relationships to one another. Creating the diagram with the programmer early in the process will speed development by allowing the programmer to begin developing and testing the sequencing rules using generic sample content in a skeleton while you continue to design and develop the actual content. The diagram you create should be accompanied by a clearly defined set of sequencing rules. Refer to *Section 7–Sequencing your SCORM Content* to learn more about writing sequencing rules.

The design activity ends in a storyboard (also referred to as a blueprint) for future development. The results of the design phase should be peer reviewed for completeness and adherence to the established design processes and procedures.

Consideration

When designing SCORM content, consider the following:

- How will you maximize the potential for content to be redeployed, rearranged, repurposed, and rewritten?
- Will SCOs cover a single learning objective or multiple learning objectives or will it vary as needed for the project?
- Will SCOs include an embedded assessment, or will the assessment be a separate SCO?
- How will SCOs be divided, structured, and sequenced?
- What media types will be incorporated?
- What other organizational policies and practices (e.g., Section 508) must be addressed?
- When, where, and how will you collect evaluation data (e.g., per SCO, aggregation, content package)?
- How many navigation options will be provided in the SCO versus the standard navigation options provided by a typical LMS?
- What colors and layouts will work best in the target LMS and in other LMSs?
- Will the use of templates and cascading style sheets facilitate rearranging, repurposing, and rewriting the content?

4.2.3   Development Phase

The development phase focuses on the creation and assembly of the content that was identified in the design phase. SCORM does not specify the tools with which you develop your content. You can develop SCORM content in nearly any authoring tool, programming language, or system that outputs electronic content. An authoring tool is software that allows you to create, rearrange, repurpose, and rewrite learning content. Some examples of authoring tools and languages include Adobe® Dreamweaver®, Adobe® Flash®, Hypertext Markup Language (HTML), JavaScript, and Extensible Markup Language (XML). At the same time content authors are building the content, the programmer should use the content structure diagram to program the sequencing rules in a functioning prototype or skeleton, with generic content. This will allow multiple iterations to test the sequencing rules before the actual content development is completed. This method will help to ensure that the sequencing rules will work properly when the content is ready to be deployed. During the development phase, follow a comprehensive quality assurance process to ensure that the content is error-free, accurate, functional, and compliant.

Key Point

Programmers should create functioning prototypes or skeletons so they can test sequencing rules at the same time the actual content is being developed. This approach can help prevent schedule delays.

The development activity ends with creating the assembled SCOs into a content package with a manifest. The results of the development phase should be peer reviewed for completeness and adherence to the design specifications.

4.2.4   Implementation Phase

The implementation phase focuses on deploying the content in a testing area on the target LMS to ensure that learner data is being recorded as you intended. After successful testing, upload the content package to the production area of the LMS and have beta testers from your target audience run the content and note any discrepancies. Refer to *Section 9–Testing Your SCORM Content* to learn more about SCORM testing practices.

The implementation phase ends when you deploy your content to actual learners via the LMS. If you are developing content for a DoD organization, you must register it in the ADL Registry and store your content packages and SCOs in a searchable content repository. Refer to *Section 10–Creating and Registering Content Metadata* to learn more about DoD requirements for the registration of metadata for your content.

4.2.5   *Evaluation Phase*

The evaluation phase collects feedback to improve training content and maximize learning. SCORM imposes no new requirements on this phase. Collecting both formative and summative data throughout all phases allows the instructional designer to see where content may need to be revised to meet the business goals, satisfy the learners, and achieve learning results. Plan to evaluate your SCORM content at multiple phases throughout your content development lifecycle using learner and client surveys and interviews. Use peer, client, and learner feedback to assess any flaws in the training materials or the development of those materials.

5   Understanding SCORM Data Model Elements

You will learn about collecting and storing data about learner performance in this section. Understanding the types of data that can be communicated via the SCORM data model enables you to tell your programmers what information you need to retrieve from or store in the LMS.

In SCORM 2004, every LMS must implement certain functionality to ensure interoperability and achieve SCORM-compliance. One element of this functionality governs how data is collected and stored. The SCORM data model elements are the only interoperable way to collect data about your learners' performance in, and interaction with, the instructional content.

5.1   SCORM Data Model Elements Overview

The SCORM data model elements are collected and stored in an LMS. A SCO must initiate all communication with the LMS. After the SCO has initiated communication, the SCO can store information on the LMS or ask the LMS to report previously stored values. The SCORM data model elements, described in detail in the SCORM RTE book, facilitate the collection of learner information as learners progress through a SCO. An LMS is required to support all of the data model elements, but you are not required to use any of them. As an ISD, you need to know what data can be communicated via the SCORM data model so you can tell your programmers what you want your SCOs to retrieve ("Get.Value") from the LMS and store ("Set.Value") in the LMS. "Value" refers to the data model element's actual *cmi* name.

> **Key Point**
>
> The SCORM data model elements define the types of data that are stored in an LMS. You must program the SCO with the correct JavaScript code to convey all of the types of information you want to send from the SCO to the LMS. The more information you want to communicate, the more programming each SCO will require.

For example, you might want to retrieve information from the LMS to customize your learners' experience such as:

- The learner's name for use inside the content (i.e., "Well done, Jane.")
- The last location in the content the learner viewed (i.e., "Do you want to start where you left off?")
- The learner's language, presentation, or other preferences

You may also want to store information in the LMS such as the learner's:
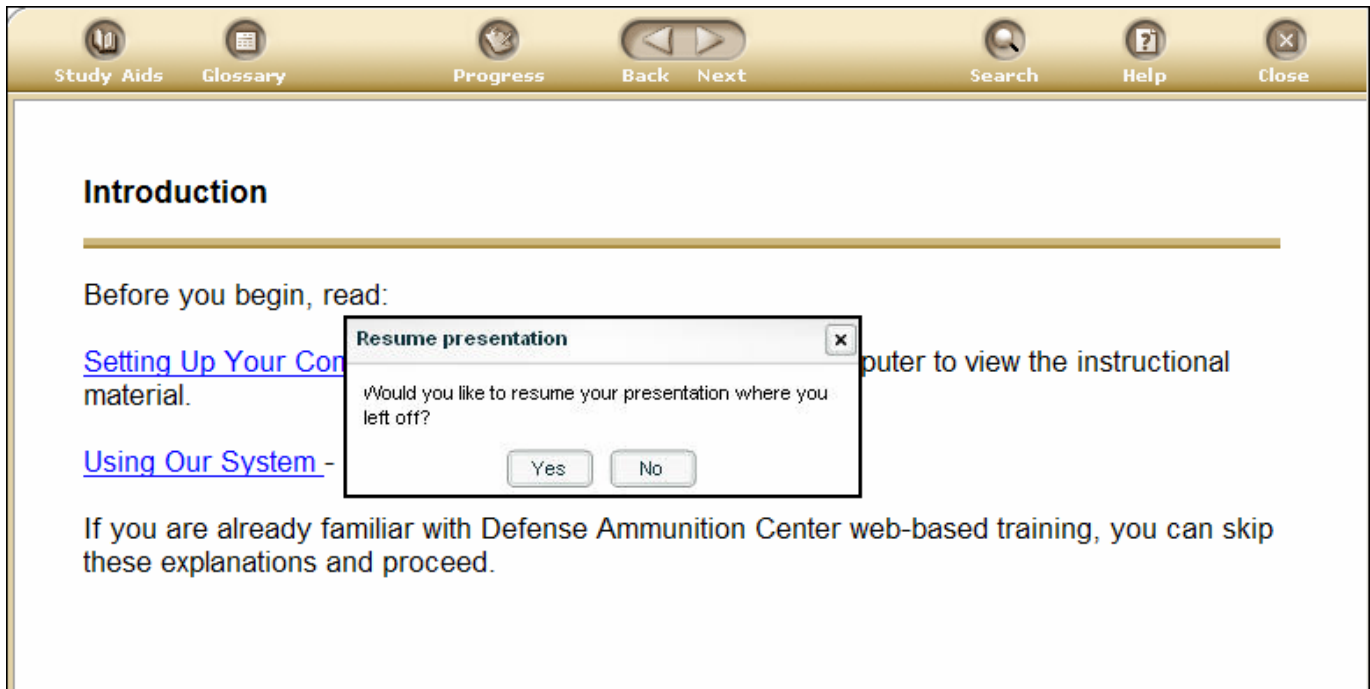
- Score
- Total time spent in a SCO
- Time spent in a single session of a SCO
- Completion status
- Responses to assessment items
- Interactions within a SCO
- Pass/fail status

Figure 5.1 Example Use of *cmi.location* Data Model Element in a SCO depicts an example of the *cmi.location* data model element in a SCO.

Figure 5.1 Example Use of *cmi.location* Data Model Element in a SCO

Before deciding which data model elements to use, you need to determine your data reporting requirements. Programming your SCOs to collect this data is time consuming and complicated, so be sure you are only collecting data that will be used, and not just collecting data for the sake of collecting data. Ask yourself these questions:

- What information about learner performance must be collected and reported?
- How will the information that you collect be used in the future?
- Are there any restrictions about the data that can be collected and how it is used?

5.2   SCORM Data Model Element Descriptions

Table 5.1 describes and provides a description and application of the SCORM data model elements using the simple name and the *cmi* name. The elements listed in this table define the only information you can collect in an interoperable manner. If you require additional data not specified in the SCORM data model elements, the data will not be interoperable.

Table 5.1 SCORM Data Model Elements

Data Model Element with *cmi* Name	Description	Application
Technical Data		
Credit *cmi.credit*	Indicates if the learner will receive credit for performance in the SCO.	Learners can take a SCO for credit or no credit. The default value is credit.
Entry *cmi.entry*	Indicates whether the learner has previously accessed the SCO so the run-time environment will know if data for the SCO exists or not.	When learners enter a SCO for the first time, the element is set to ab-initio. If the learner is re-entering a suspended session, the element is set to resume.
Launch Data *cmi.launch_data*	Provides data specific to a SCO that the SCO can use for initialization.	Allows SCOs to be configured with data from the LMS at the time of launch. For example, importing new statistical data into a SCO on statistics.
Location *cmi.location*	Represents a location in the SCO.	Used for book-marking the learner's position in a SCO in a given instance, allowing the learner to resume the SCO at the same point at which learning was suspended.
Mode *cmi.mode*	Identifies how the SCO may be presented to the learner.	Browse is typically used when learners do not want information about their performance to be stored. Normal is used when learners want to store information about their performance. Review is used by learners who have already completed the SCO but want to view the item again as a refresher.
Suspend Data *cmi.suspend_data*	Provides additional space to store and retrieve data between learner sessions.	If learners start the SCO, but do not complete it, the state of suspension (where the learner stopped) is tracked.
Content Initialization		
Completion Threshold *cmi.completion_threshold*	Identifies a value for comparison with the learner's progress in a SCO to determine if the SCO should be considered complete.	The completion threshold, determined by the ISD, allows a SCO to set its status when something happens (i.e., a specific number of pages have been viewed).
Learner ID *cmi.learner_id*	Identifies the learner on behalf of whom the SCO was launched.	This is a unique alpha-numeric code that refers to a single user. It allows the LMS to associate data with a specific learner.

Data Model Element with *cmi* Name	Description	Application
Learner Name *cmi.learner_name*	Allows the SCO to present the name of the learner inside the content.	Typically used to customize learning content, for example: "Welcome back Sgt Thomas." or "Nice work, Capt. Brown."
Learner Preference *cmi.learner_preference*	Specifies preferences associated with the learner's use of the SCO.	Can be used to present learner-specific content within a single SCO. If there are multiple languages in a single SCO, this data model element would select the learner's preferred learning language. If the SCO is Section 508-compliant, it can select the audio level, delivery speed, or audio captioning.
Maximum Time Allowed *cmi.max_time_allowed*	Indicates the amount of accumulated time the learner is allowed to use a SCO in an instance.	The learner attempt begins with the beginning of the first session and continues until the activity terminates.
Time Limit Action *cmi.time_limit_action*	Indicates what the SCO should do when the maximum time allowed is exceeded.	If the learner exceeds the prescribed time limit, the ISD can choose to either automatically exit the SCO or allow the learner to continue. A standard message such as, "Your session has ended because you have exceeded the time limit" may also be included.
Score Reporting		
Completion Status *cmi.completion_status*	Indicates if the learner has completed the SCO.	The completion status, determined by the ISD, can be a test score, navigation through content, completion activities, etc.
Interactions *cmi.interactions*	Describes a collection of learner responses, such as responses to questions or tasks for the purpose of measurement or assessment.	Frequently used in tests or quizzes to collect learner response information. Refer to *Section 6–Using Assessments in SCORM Content*, for more information on *cmi.interactions*.
Objectives *cmi.objectives*	Specifies learning or performance objectives associated with a SCO.	May be used to impact sequencing decisions or how you divide your content.
Progress Measure *cmi.progress_measure*	Identifies how much progress the learner has made toward completing the SCO.	The progress can be based on the completion of a certain number of objectives related to the SCO, the number of pages presented to learners, etc.
Scaled Passing Score *cmi.scaled_passing_score*	Identifies the scaled passing score required to master the SCO.	Should be set to your minimum passing score.
Score *cmi.score*	Identifies the learner's score for the SCO. A SCO can only report one score.	This is typically the result of some interaction the learner has with the content.

Data Model Element with *cmi* Name	Description	Application
Success Status *cmi.success_status*	Indicates if the learner has mastered the SCO.	The criteria, defined by the ISD, can be based on a percentage of interactions being passed or objectives being met, a score for a test or quiz, etc.
Comments		
Comments From Learner *cmi.comments_from_learner*	Enables the collection of comment text, the location the comment was made, and the timestamp of when the comment was made.	Learners can provide text feedback (comments, suggestions) about the SCO or their learning experience, while in the content, that can later be used to update or improve the learning experience.
Comments From LMS *cmi.comments_from_lms*	Contains comments and annotations intended to be made available to all learners.	Instructors can provide information intended for all learners viewing the content, such as a clarification of an issue or information between updates to the content.
Exit Data		
Exit *cmi.exit*	Indicates how or why the learner left the SCO.	Can be used to impact sequencing decisions.
Session Time *cmi.session_time*	Identifies the amount of time that the learner has spent in the current instance of the SCO.	The ISD defines the value and meaning of session time. The LMS uses this time to compute total time.
Total Time *cmi.total_time*	States the learner's cumulative time for all sessions of a specific SCO for a given learner attempt.	Stores the total time spent in every session of a given SCO, for a given learner.

This page intentionally left blank.

6   Using Assessments in SCORM Content

You will learn about structuring SCORM assessments as single or multiple SCOs and get an overview of test banks and _cmi.interactions_ in this section. You can collect metrics for evaluation reporting and significantly improve learner performance by matching performance on individual learning objectives with remediation and feedback strategies to further customize learning experiences.

SCORM does not directly address how to design and develop assessments nor when and how a SCO should be considered an assessment. As a result, many organizations have

> Information in this section was derived from the Carnegie Mellon _Best Practices for the Design and Development of SCORM Assessments_ guide.

resorted to delivering assessments using proprietary assessment solutions provided by their LMS. This is not the ideal solution because assessments and assessment metrics defined within a given LMS solution are not interoperable or reusable in other LMSs.

6.1   Structuring SCORM Assessments

SCORM assessments can be structured in two ways: as a single SCO containing multiple test items or as multiple SCOs, each of which contains a single test item. Each assessment SCO must report a score and completion status, regardless of the way it is structured.

6.1.1   Assessments as a Single SCO

When you present an assessment to learners as a single SCO, the LMS receives a comprehensive score, such as 92% or 75% and a completion status is set. If you need to collect detailed response information for each test item in an interoperable way, you can use the _cmi.interactions_ data model element. This facilitates the formative and summative evaluation process because you can report learner behaviors and assessment validity and reliability at a granular level. Since the assessment is a single SCO, it preserves the integrity of learners' testing experiences. You can present one test item per screen or all test items on a single screen. You can also control randomization of the test items inside the SCO, and you don't have to write as many sequencing rules to control the presentation and remediation behaviors of each test item.

This level of functionality within a SCO requires more time to program, but ISDs will have to design fewer sequencing rules and programmers will have to code fewer sequencing rules. The benefit to learners with this type of strategy is a better testing experience because there is less processing time between the delivery of each test item. Figure 6.1 depicts how one SCO equates to one assessment.

Figure 6.1 Assessment Structure with One SCO Equal to One Assessment

6.1.2   Assessments as Multiple SCOs

When you present an assessment to learners as multiple SCOs—a series of SCOs, each containing one test item—the LMS can receive a score and a completion status for each test item. Using *cmi.interactions*, you can collect detailed response information for each test item in an interoperable way. This facilitates the formative and summative evaluation process, because you can report learner behaviors and assessment validity and reliability at a granular level. To move between test items, you can use sequencing rules. To compile a comprehensive score, you can use the roll-up function in sequencing, but roll-up is complicated and requires programming.

Unfortunately, delivering an assessment as multiple SCOs can create an interruption in the "test flow" for learners. The LMS must evaluate each SCO individually, and some LMSs may require learners to manually submit each SCO when completed. The processing time could create a slight delay after each test item as the LMS reviews the sequencing rules and determines which item to deliver next. Learners may also have to navigate between the LMS and each SCO. These factors can cause your learners to lose concentration during the assessment, which is not optimal during assessment delivery. Figure 6.2 depicts how one SCO equates to one test item.

Figure 6.2 Assessment Structure with One SCO Equal to One Test Item
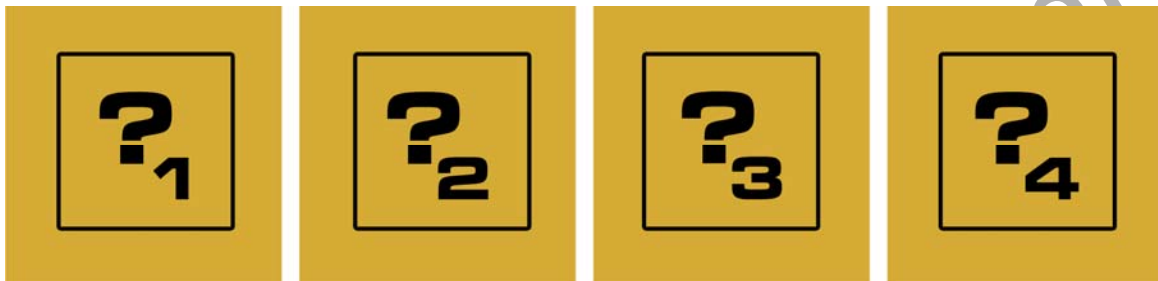
6.2   Creating Test Banks

A test bank reduces the likelihood of two different learners receiving the same assessment, and ensures that learners do not receive the same assessment on multiple attempts. A test bank is a collection of related test items. Typically, a test bank delivers a given number of items that are pulled randomly from the test bank to test a given learning objective or group of learning objectives. All of the functionality for test banks must be programmed into your SCO(s) in advance since SCORM does not specify how to structure or deliver assessments.

When you create test banks for SCORM assessments, you can either create a single SCO as a test bank for a given learning objective (in which case you would require multiple SCOs for a complete assessment) or you can create a single SCO made up of multiple test banks.

Based on the limitations described in Subsection 6.1.1 above, for structuring each assessment item as a single SCO, the most effective way to structure your assessment would be with a single SCO comprised of multiple test item banks. Using *cmi.interactions*, you will still be able to collect all of the data you require for each item, and you will preserve the flow of the learners' experience as they proceed through the assessment. Programming the functionality of the *cmi.interactions* data element is very complex, so you should identify your data collection requirements in advance. Figure 6.3 depicts how one SCO equates to one test bank and Figure 6.4 depicts how one SCO equates to multiple test banks.

Figure 6.3 Assessment Structure with One SCO Equal to One Test Bank
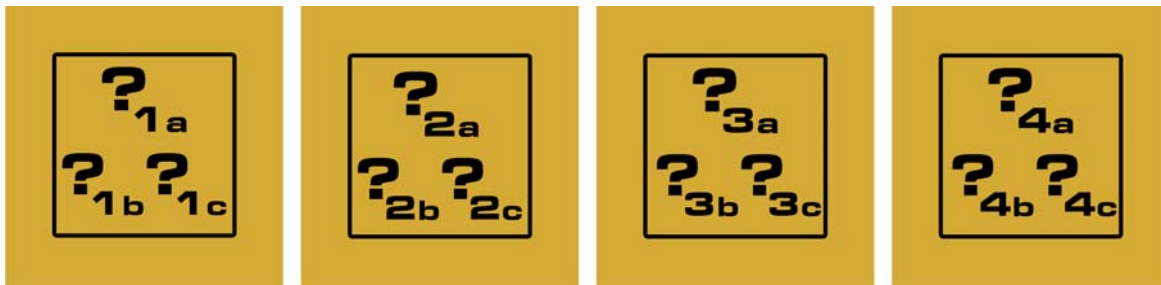
Figure 6.4 Assessment Structure with One SCO Containing Multiple Test Banks
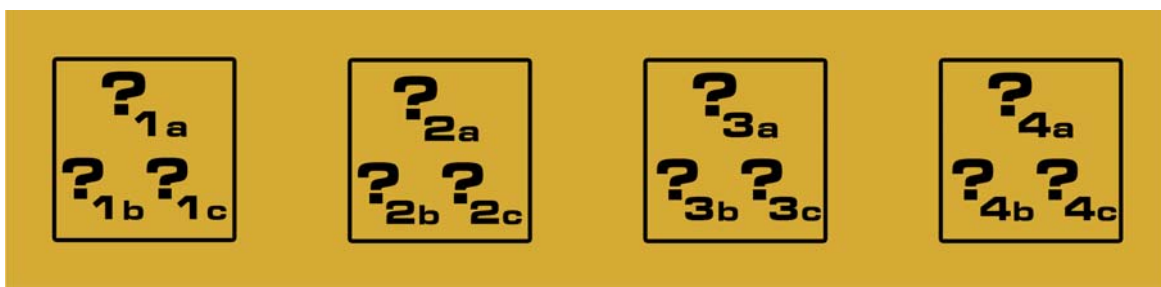
6.3 Identifying Data Collection Requirements

The amount of data you can collect from an assessment can be significant, so balancing the amount of programming required to collect the data against the way you plan to use the data is essential in SCORM assessments. Typically, in the analysis phase, you identify the types of data you need to collect and report about the learner's experience. Deciding to collect all of the data you possibly can about learners' performance on an assessment can add countless hours of programming to your project. So, while you are gathering requirements in the analysis phase, ask your team and/or client about their data collection needs for the project, including:

- Who will have access to the data?
  - What security considerations are there for the data if it is reported individually?
- How will the data you collect be used?
  - Will the data be used to improve the content and the assessment?
  - Will the data be used to identify individual learners who excel or perform poorly?
- What stage of the evaluation process are you in, formative or summative?
  - What are the contractual reporting requirements for the project in this evaluation stage?
  - How must the reported data be delivered, as aggregated or individual data?
- What type of reporting is provided by your LMS to facilitate data reporting?
  - How many report options are provided?
  - Are the reports easy to run and automatically formatted?

The most common data collected from assessments is addressed by the SCORM *cmi.interactions* data model element.

6.4   Understanding *cmi.interactions*

The *cmi.interactions* SCORM data model element allows a SCO to send data to the LMS about a learner's performance on an assessment in an interoperable way. It provides a detailed model for designers and programmers to collect metrics about learner response or performance within a SCO, particularly data related to performance on assessments such as correct response, learner's response, duration taken to respond, and weight of the particular item relative to the overall assessment score.

With *cmi.interactions*, instructional designers can collect metrics for formative and summative evaluation reporting, and link *cmi.interactions* and other *cmi* data model elements to sequencing rules to create remediation or adaptive learning strategies that further customize learning experiences. This may significantly improve learner performance by allowing designers to better match performance on individual learning objectives with remediation and feedback strategies resulting in more individualized learning. For example, a learner might incorrectly select "A" on a multiple choice question where the correct response is actually "C". Selecting "A" as the incorrect answer could indicate that the learner has certain misconceptions about a specific concept. Using *cmi.interactions* in combination with other data model elements, you could individualize the learning by sending the learner to a specific SCO that addresses the weakness identified during the initial assessment. You could then reassess them, and, based on their results, assign further remediation or return them to their original learning path.

A single interaction describes a test item. Each interaction can store:

- Question identifier—used to associate the question to a database or master list of questions

- Type of question—multiple-choice, true/false, matching, etc.

- Order of the responses as they were presented to the learner—if the order of the responses was randomized

- Correct answer—what the learner should have answered

- Learner's response—what the learner actually answered

- Whether the learner's response was correct or not

⊕ **Consideration**

You must determine in advance exactly what type and how much information you want to store, and ensure that your SCO is programmed correctly to store it in the LMS and then convey the actual reporting needs to the programmer. Collecting too much data about assessments:

- Increases programming time (may increase cost of the project).

- Increases the amount of communication between the SCO and the LMS (may slow LMS response time).

- Increases the storage requirements of the LMS (may reduce amount of total storage available).

- May violate the rights of learners in some cases (particularly in situations where learners are union employees).

7   Sequencing Your SCORM Content

You will learn about structuring your content for sequencing, including the origins of sequencing, variables, control modes, and rule conditions in this section. Sequencing gives designers the ability to create individualized instruction based on learners' performance and prior knowledge.

In traditional multimedia and CBT, branching enabled (or sometimes forced) learners to move from one piece of content to another relatively seamlessly. If learners failed to understand a certain concept, they could be remediated to new content or review existing content. Learners may or may not have known they were moving from one lesson or module to another. This was possible because robust authoring and delivery systems gave designers and developers nearly limitless programming options for structuring and branching their content. The functionality was hard-coded, whether based on a linear or an adaptive model. If you tried to remove one piece of content from the whole, the branching rules would break. Figure 7.1 shows content delivered as a traditional CBT with branching between each learning objective (white box).

> Information in this section was derived from Section 7 of the Carnegie Mellon *SCORM Best Practices Guide for Content Developers*.

Figure 7.1 Depiction of a Traditional CBT Course with Internal Branching
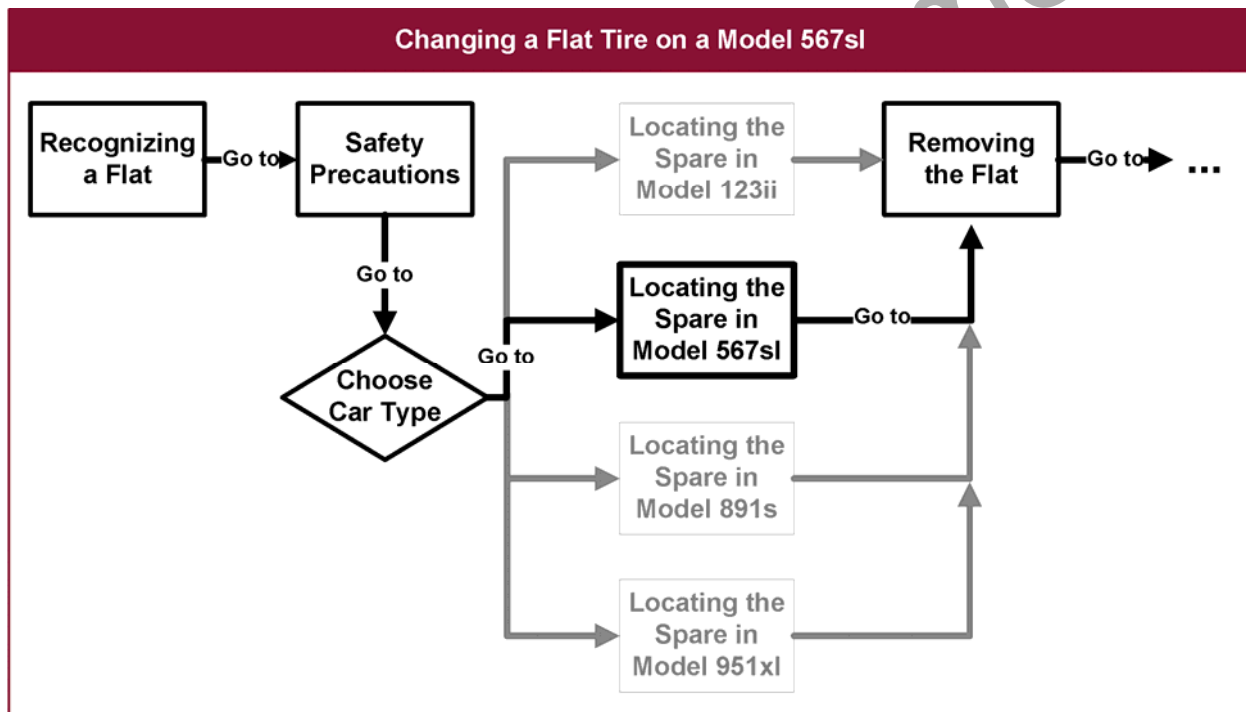
7.1   Origins of Sequencing in SCORM

In the early versions of SCORM, a set of SCOs was presented to learners, typically via a table of contents, and learners could select the SCOs they wanted to see in any order. Some instructional designers found this aspect of SCORM frustrating, since they were being asked to design granular SCOs, but with granular SCOs they were unable to ensure that learners received the SCOs in the order they prescribed. This was particularly troubling for procedural content that required learners to experience the content in the order they would perform the procedure.

Figure 7.2 depicts individual SCOs (yellow boxes) delivered as a collection of SCOs in a SCORM 1.2 content package with no relationships between them.

Remember that all of the diagrams in this Guide are color- and shape-coded so that you can quickly identify the difference between SCORM components. Yellow boxes represent SCOs. The green boxes represent aggregations. The red boxes represent a root aggregation or organization.

Figure 7.2 Depiction of a Multi-SCO SCORM 1.2 Content Package
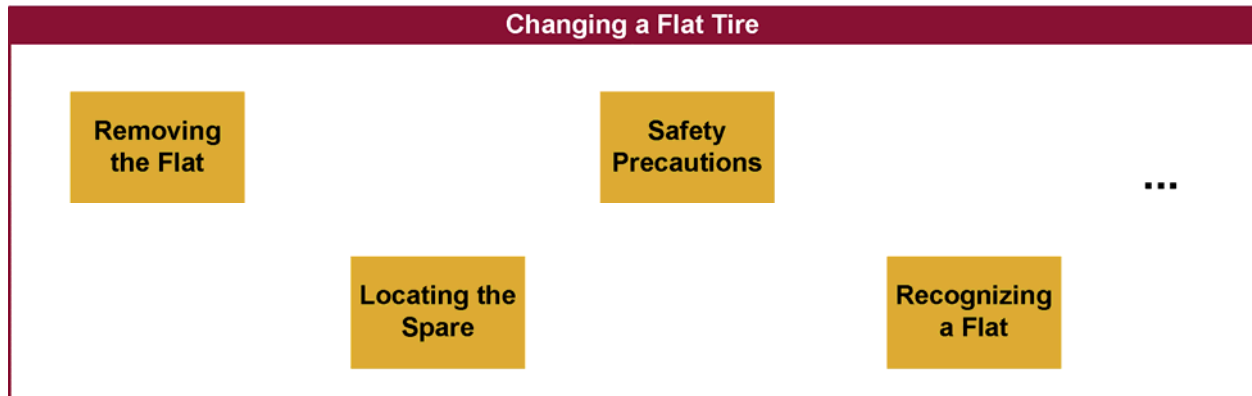
To overcome this limitation, some ISDs created very large SCOs that had internal branching. This enabled them to comply with SCORM without sacrificing control over the learners' experiences. However, these SCOs were typically too big to be reusable, which violated the spirit of SCORM. Other ISDs used the proprietary sequencing functionality provided by their LMS, which meant that the sequencing was lost when the content package was moved to another LMS. Figure 7.3 depicts a single SCO (yellow box) delivered in a SCORM 1.2 content package. This type of SCO used internal branching to move between portions of the instruction.

Figure 7.3 Depiction of a Single SCO with Internal Branching in a Content Package

Since ADL promotes interoperability and reusability, hard-coding functionality within or between SCOs contradicts the ADL architectural functionality. Content was often not interoperable when hard-coded sequencing rules were present or when sequencing rules were defined using one LMS's proprietary functionality because the sequencing functionality of one LMS could not be processed by another LMS. Likewise, content could not be reused when individual SCOs relied directly on the presence of other SCOs. Hard-coding also limits the ability to create new or custom content structures from the same instructional materials, since each time a new structure is desired, the code attached to each individual SCO has to be revised.

7.2   Sequencing versus Branching

With sequencing in SCORM 2004, ISDs can prescribe the manner in which learners receive individual pieces of content from the LMS interoperably. As a result, you can design more granular content and allow the LMS to control the movement of learners from SCO to SCO in accordance with the behaviors you specify; this increases the possibility that your SCOs will be reusable because there are no hard-coded rules in your SCOs to modify. Sequencing can also be used with highly granular reusable content to add context to your content when you have context-neutral SCOs. Refer to *Section 2–Designing Reusable Content*, for more information.

This document refers to both sequencing and branching. "Branching" means the intra-SCO, hard-coded rules, that occur inside an individual SCO. It is not tied to the LMS or to the content package, so it is outside the scope of SCORM. Table 7.1 compares branching and sequencing.

Table 7.1 Comparison of Branching and Sequencing Capabilities

Feature	Branching	Sequencing
Enables learners to view content as prescribed by ISD	✓	✓
Allows adaptive movement through content	✓	✓
Is hard-coded into the content	✓	
Allows content to remain in small, reusable pieces		✓
Uses standard rules that any SCORM 2004-compliant LMS can read from the manifest		✓

Sequencing consists of a set of standardized behaviors defined in the SCORM Sequencing and Navigation book that all SCORM 2004-compliant LMSs must support. Sequencing allows you to control what content is presented to learners and when it is presented in an interoperable way.

Sequencing rules reside in the manifest file, a part of the SCORM content package, so content can be sequenced in an interoperable manner, without relying on hard-coded data inside a SCO. Figure 7.4 depicts a process commonly referred to as the sequencing loop; the process starts with sequence and progresses clockwise. Conceptually, the LMS reads the sequencing rules from the manifest file, locates the appropriate SCO to deliver to the learner, launches the SCO to the learner's browser, collects data about learners' performance and status, and then processes the next set of sequencing rules from the manifest.

Figure 7.4 Sequencing Loop
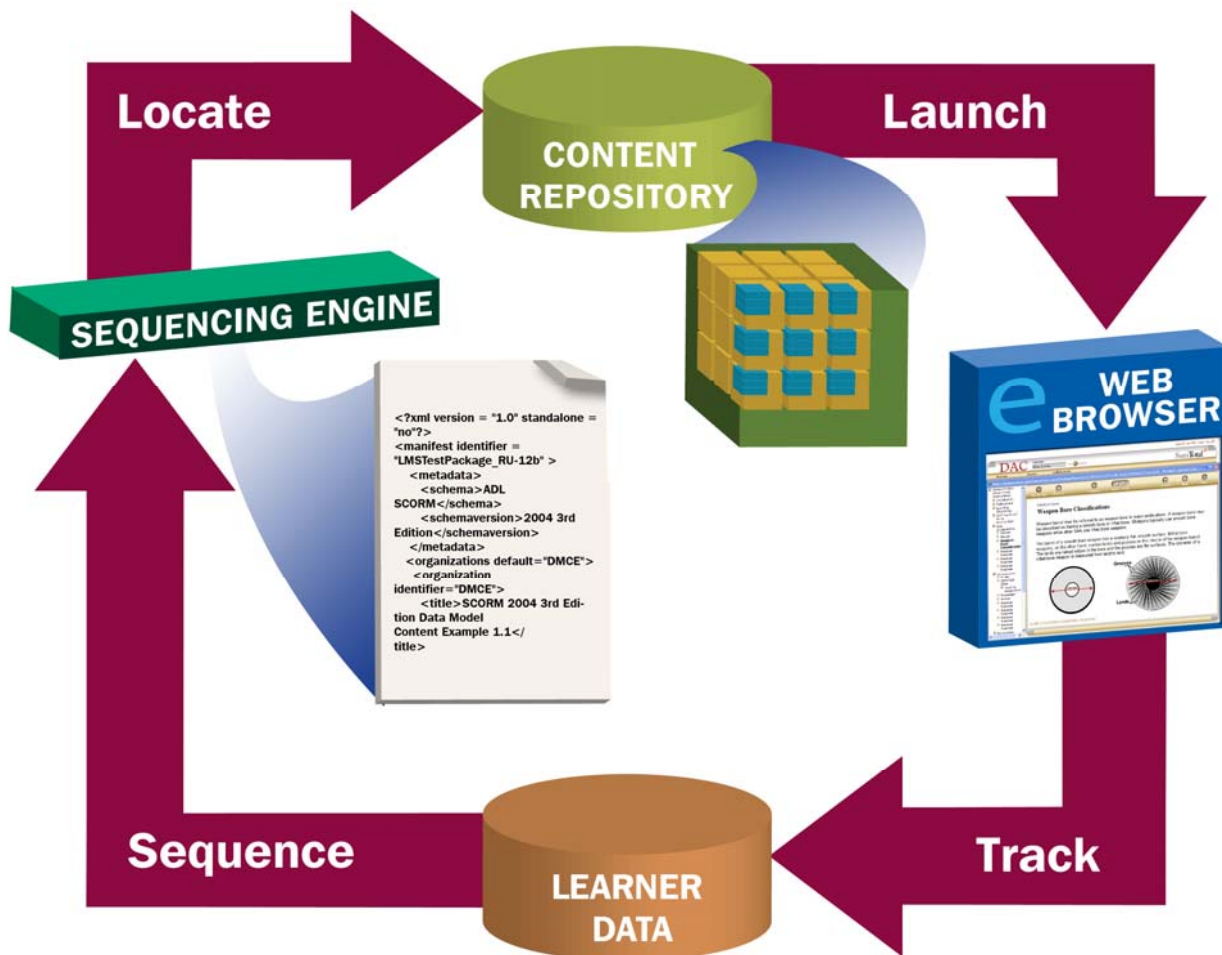
7.3   Determining Your Sequencing Requirements

Before you can structure your content for sequencing, you need to determine what your assessment and remediation requirements are and how you want learners to experience the content. Will learners be able to choose the content they see, or will you prescribe the order in which they see it? Do you want to adapt the learner's experience based on their choices or decisions? How will you determine when the learner has completed the content? After you know what your sequencing requirements are, you can begin to structure your content and define the sequencing behaviors you need.

7.4   Structuring Your Content for Sequencing

The sequencing in SCORM is based on a hierarchical tree structure, so specifying the actions and behaviors you want for your learner requires the creation of a content structure diagram (CSD). The CSD depicts the relationships among your content, allowing you to organize your SCOs into aggregations. If you have created a traditional outline for your content, you can use that to start your CSD since it already provides a hierarchical tree-based structure. For example, the content structure outline might look like:

Transportation of Hazardous Materials HAZMAT

- Types of HazMat
- Hazard Classes
  - Hazard Class 1 Explosives
  - Hazard Class 2 Gases
  - etc.
- Transportation Documentation

Figure 7.5 depicts a partial content structure diagram created from the content structure outline above.

Figure 7.5 Partial Content Structure Diagram
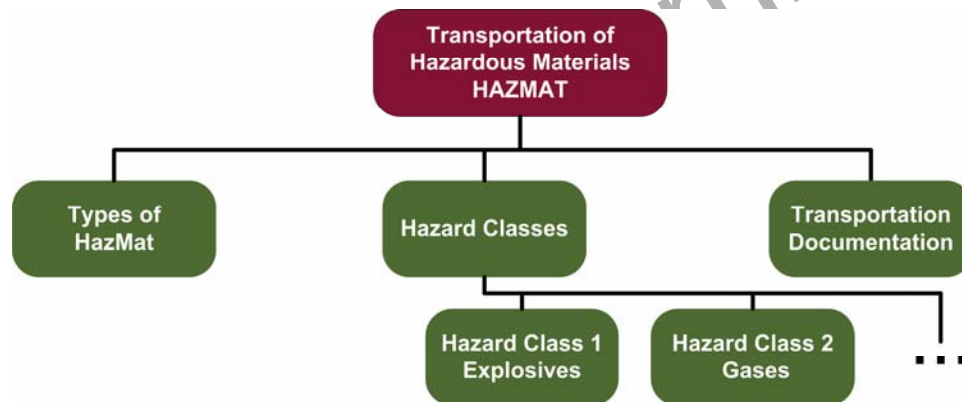
Each of your SCOs should fit into the initial structure you create, so the expanded outline below will become Figure 7.6, a partial content structure diagram with SCOs. The content structure diagram shows how SCOs are grouped to fit into the structure. These groups, represented by the green boxes, are *aggregations*.

Transportation of Hazardous Materials HAZMAT

Types of HazMat

- Types of HAZMAT Overview
- Hazardous Waste
- Marine Pollutant
- Hazardous Substance
- Elevated Temperature Material

Hazard Classes

- Hazard Class 1 Explosives
  - Division 1.1 Mass Explosion Hazard
  - Division 1.2 Fragmentation Hazard
  - Division 1.3 Fire Hazard
  - Division 1.4 Minor Explosive Hazard
  - Division 1.5 Very Insensitive Explosive Hazard
  - Division 1.6 Extremely Insensitive Explosive Hazard
- Hazard Class 2 Gases
- etc.
- Transportation Documentation

Figure 7.6 Partial Content Structure Diagram With SCOs
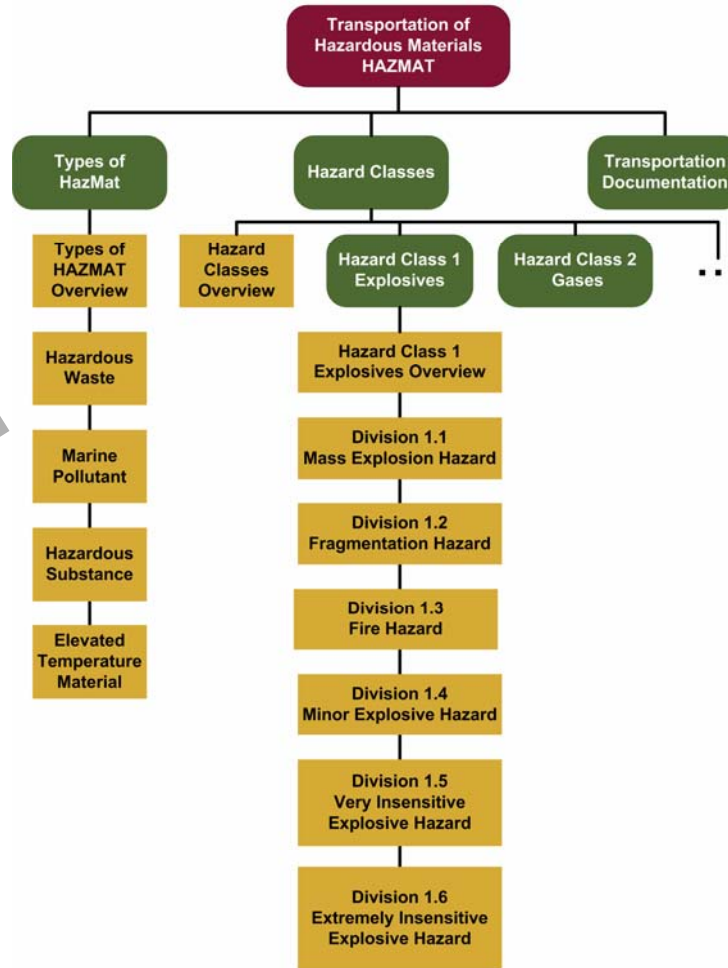
7.5   Defining Sequencing Rules

Since the sequencing of your SCOs is controlled by the LMS, you must carefully describe the actions and behaviors you desire for each SCO and aggregation, all the way back to the root aggregation. If you fail to do this, the actions and behaviors of your content will be the default values defined by SCORM, which may not result in the type of learning experience you had planned or desired. The sequencing rules (generated by the programmer in the manifest) will enable the behaviors you describe for your content structure diagram. You will need to work closely with the programmers to ensure you clearly define what you intend for learners to see and experience. The programmer will probably require multiple iterations and a significant amount of testing to create the exact experience you define.

The information in the following subsections is fairly detailed. It may help you to better understand how to communicate with your programmer and how to structure your content, but you'll need to rely heavily on your programmer's expertise to obtain the full benefits of sequencing's capabilities.

7.5.1   Control Modes

As stated previously, there was no way to control the order SCOs were viewed in SCORM 1.2—the learner could choose any SCO at any time. In SCORM 2004, sequencing gives you the option to control the order in which the SCOs in a given aggregation are viewed. If you do not need to implement sequencing, the default setting will allow learners to choose any SCO at any time.

However, if you need to control the order in which your learners experience the SCOs within a given aggregation, sequencing provides several modes of control that you set to TRUE or FALSE. The following list includes the most common control modes.

Control Mode	Description
Choice	Allows learners to select the order in which they view the content.
Flow	Requires learners to view the content in an order defined by the instructional designer. Next and previous buttons may be provided by the LMS.
Forward Only	Prevents learners from going back to an activity within a given aggregation once they've started another activity.

For example, if you wrote no rules for the sequencing structure in Figure 7.7, the learner could choose to see Removing the Flat then Safety Precautions, and so on. However, if you wanted the learner to view each SCO in order, you could set FLOW=TRUE and the learner would have to complete Recognizing a Flat followed by Safety Precautions, and so on.

Figure 7.7 Basic Sequencing Structure
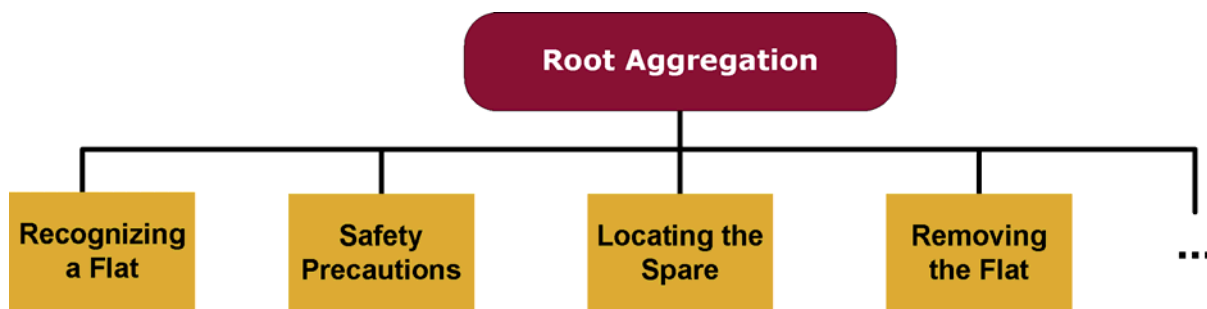
You can also set combinations of sequencing rules, but be careful, as that could create problems. For example, when specifying the control mode for your learners it is possible to say CHOICE=TRUE and FORWARD ONLY=TRUE. In Figure 7.7, if you set the control mode to CHOICE=TRUE and FORWARD ONLY=TRUE and a learner starts with Removing the Flat, the previous SCOs (Recognizing a Flat, Safety Precautions, and Locating the Spare) will not be available.

7.5.2   *Parents and Children*

In sequencing, a SCO is always considered a child—it cannot be a parent and have a SCO beneath it in the content structure. Aggregations, however, can be both parents and children. Remember that aggregations are simply collections of SCOs and other aggregations. An easy way to remember how rules apply in sequencing is that all of the children must follow all of the rules of the parent and no child is special. There is no inheritance, so the rules set at one parent level apply only to that parent's immediate children. Any time you need to define a special rule for a SCO or aggregation, you need to create a new parent for it. For example, Figure 7.7, above, depicts a series of SCOs under a root aggregation. Each of the SCOs in this root aggregation would follow the set of rules defined for the root aggregation, so you could allow learners to either choose to see the SCOs in any order (CHOICE =TRUE), or you could force them to see them in order (FLOW=TRUE).

However, if you wanted to allow learners to choose either Recognizing a Flat or Safety Precautions before going on to the other SCOs, you would have to create a new parent for them. Figure 7.8 depicts the same content with a parent aggregation called Safety added above the Recognizing a Flat and Safety Precautions SCOs. Rules can now be assigned to the Root Aggregation that will apply only to the Safety Aggregation and the Locating the Spare, Removing the Flat and subsequent SCOs. A unique set of rules can now be added to the Safety Aggregation.

Figure 7.8 Basic Sequencing Structure with One Aggregation
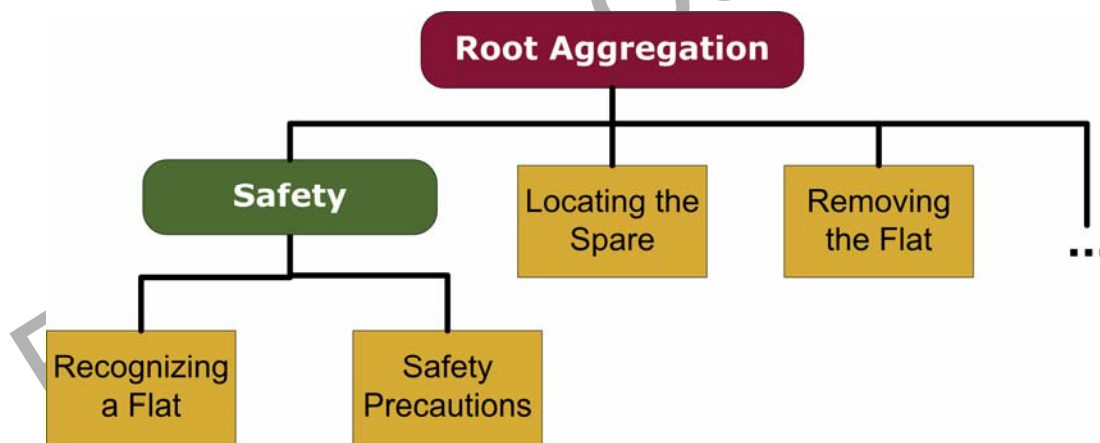
So for Figure 7.8, you might write the following rules for control mode:

- For the Root Aggregation, learners must view all of the items in order. They can go back at any time.
- For the Safety Aggregation, allow learners to choose either Safety Precautions or Recognizing a Flat.

7.5.3   Pre- and Post-Condition Rules

Since sequencing rules are not hard-coded into the SCOs, some rules need to be processed by the LMS before a SCO is launched (pre-condition rules) and others need to be processed by the LMS after a SCO is launched (post-condition rules). As long as you clearly specify what you want to happen when describing your sequencing behaviors, your programmer will determine whether or not to apply a pre- or post-condition rule.

Sequencing rules follow an if/then pattern using the limited vocabulary listed in Table 7.2 If/Then Pattern of Sequencing Rules. As with control modes, your programmer will carefully select the if/then combinations to avoid creating an awkward or confusing situation for your learners.

Table 7.2 If/Then Pattern of Sequencing Rules

If the item is/has

- Satisfied
- Objective Status Known
- Objective Measure Known
- Objective Measure Greater Than
- Completed
- Progress Known
- Score Greater Than
- Score Less Than
- Attempt Limit Exceeded
- Time Limit Exceeded
- Outside Available Time Range
- Always

Then _____ BEFORE the SCO is launched.

- Skip
- Disable
- Hide from Choice
- Stop Forward Traversal

Then _____ AFTER the SCO terminates.

- Exit Parent
- Exit All
- Retry
- Continue
- Previous

7.5.4　Roll-Up Rules

Roll-up rules are used to report the status of children to their parents. Each aggregation conceptually asks its children what their status is. The aggregation can then determine its status and report its status up the tree until the root aggregation knows the status of each of its children. Like standard sequencing rules, roll-up rules follow an if/then pattern using the limited vocabulary listed in Table 7.3 If/Then Pattern of Roll-Up Rules. Roll-up rules can apply to all children, any children, none, at least "x" children, or at least "y"% of children where "x" and "y" are specified by the designer.

Table 7.3 If/Then Pattern of Roll-Up Rules

If
- Satisfied
- Completed
- Attempted
- Objective Status Known
- Objective Measure Known
- Activity Progress Known
- Attempt Limit Exceeded
- Time Limit Exceeded
- Outside Available Time Range

Then
- Satisfied
- Not Satisfied
- Completed
- Not Completed

To continue the example from Figure 7.8, so you have a complete set of sequencing behaviors described, you might add the following roll-up rules to Figure 7.9:

Figure 7.9 Basic Sequencing Structure with Roll-up Rules
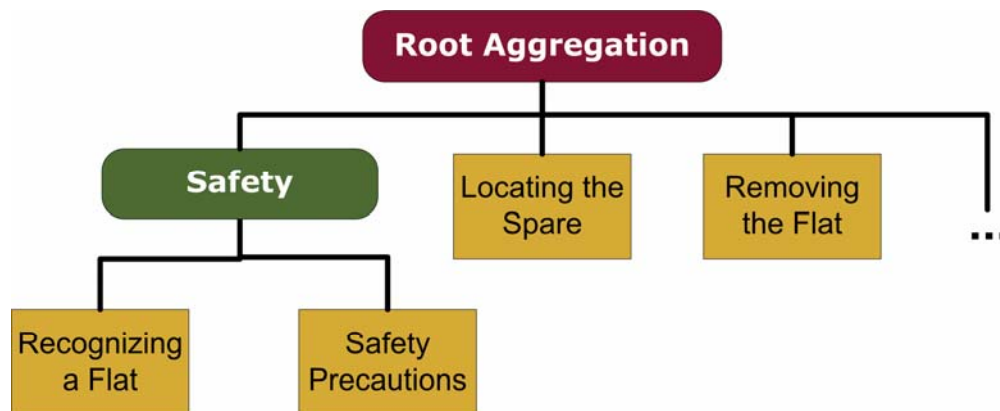
For Figure 7.9, you might write the following rules for control mode:

- For the Root Aggregation, learners must view all of the items in order. They can go back at any time. (Control Mode Rule)

- For the Safety Aggregation, allow learners to choose either Safety Precautions or Recognizing a Flat. (Control Mode Rule)

- If learners complete one of the SCOs in the Safety Aggregation, then the Safety Aggregation is complete. (Roll-up Rule)

- To complete the Root Aggregation, learners must complete the Safety Aggregation and all the SCOs in the Root Aggregation. (Roll-up Rule)

7.5.5   *Understanding Sequencing Objectives*

Some terms you may have used to signify a specific function of instruction could have different meanings in SCORM when you sequence your content. You should keep in mind the definitions of these words within the context of SCORM sequencing.

In traditional instructional design, a learning objective is used to measure the attainment of a knowledge, skill, or ability in accordance with a predefined behavior, a prescribed condition, and an achievement standard. In sequencing, the variables that can store information about one SCO in the LMS that can be retrieved for later use to impact another SCO are also called "objectives," but they have nothing to do with learning objectives. In this Guide, sequencing objectives are referred to as "variables".

Using sequencing rules and variables, the LMS can store information for later retrieval to determine which SCO should be delivered next. Each SCO can set or read multiple variables, a single variable can be set or read by multiple SCOs. When discussing your sequencing behaviors with a programmer, make sure you are both referring to objectives and variables in the same way to avoid invalid rules.

In SCORM, a sequencing variable has two values that can be used in sequencing rules: a success status and a score. You must define for the programmer the required sequencing behaviors and values needed to control the flow of the content; the programmer can then code the SCO properly. The success status can have a value of passed or failed. The score can have any decimal number between -1 and 1. For example, if you want the passing threshold to be 85%, then the score would be programmed as .85, not 85%.

This page intentionally left blank.

8   Packaging Your SCORM Content

You will learn about content package sizing and assembly, and the manifest file in this section. The content package provides a standardized, interoperable way for you to exchange digital resources between different LMSs, content repositories, and operating systems.

8.1   SCORM Content Package Overview

The content package is a self-contained .zip file that contains everything needed to deliver content to learners via a SCORM 2004-compliant LMS. The programmers on your team typically create the SCORM content package. Figure 8.1 depicts the items in your content package.

A SCORM content package contains two principal parts:

- The XML manifest file that describes
    - All of the SCOs or assets you want to include in the package
    - A representation of the content structure diagram you created (called the organization)
    - The sequencing rules
    - Metadata for the SCOs, aggregations, and the package itself
- All of the actual SCO and asset files for the content package

> **Key Point**
>
> If you are developing content for in-house use only and you never plan to run it in another LMS, you may not need to worry about packaging it since you won't be exporting it. If at a later point you are required to run your content in another system, you can generate a SCORM content package anytime.

After you have developed all of your physical SCO files and specified the sequencing behaviors you want for your content, you can prepare to package your content for SCORM delivery. When your content is ready for packaging, archive any incomplete or unused materials and ensure that your programmers can access everything they need to compile the content package. Figure 8.1 depicts a SCORM content package.

Figure 8.1 Representation of a SCORM Content Package
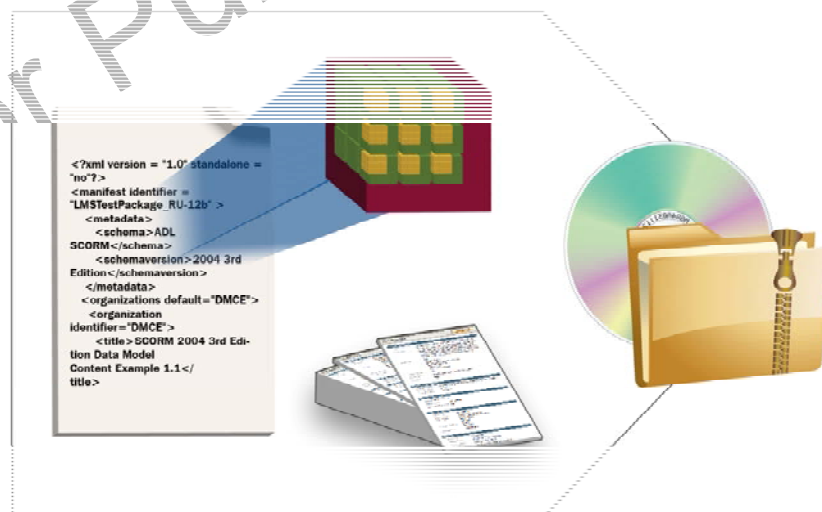
There are tools available to create content packages or your programmers may choose to create them from scratch. Some authoring tools will create the entire content package after you load your SCOs and assets into the tool. Ensure that the tools you select match the knowledge, skill, and ability levels of the team members who will use them.

8.2 The Manifest File

The content package is organized by an Extensible Markup Language (XML) file called a manifest. The manifest is a detailed, machine-readable set of instructions, structured in a manner specified by SCORM, that organizes your content package and tells the LMS when, how, and what content to deliver to your learners.

The manifest is typically created by an authoring tool, though some programmers prefer to create them from scratch using an XML editor. The manifest file is always named imsmanifest.xml and it always appears at the top level of a content package, regardless of the structure of the rest of the package.

8.3 Considerations for Content Package Size

The sequencing structures you create for your content may result in a larger or more complex manifest file. The LMS uses the manifest to know what and when to sequence.

You should check with your LMS administrator to get an understanding of any limitations your LMS may have regarding package size, number of SCOs, and performance issues related to complex sequencing. For example, a content package with 500 SCOs with complex sequencing rules may have some lag time as the LMS transitions between activities and processes sequencing rules. Likewise, if you are delivering a 200-hour qualification course, it may be difficult to structure and deliver it as a single content package. These are typically extreme cases, but learners may face delays in receiving the content.

To avoid any issues, you may want to consider chunking a single, very large content package into multiple content packages. If you need to control the order of delivery of your content packages, you could use the prerequisite functionality provided by your LMS. However, that functionality is not specified by SCORM because SCORM does not define the behaviors occurring outside a content package.

Content package size will depend on the:

- Structure of the learning experience.
- Number and complexity of sequencing rules.
- Number of SCO files.

8.4 Assembling the Content Package

After the programmers have all of the necessary files, they can create the manifest with your organization or root aggregation and sequencing rules. A completed SCORM content package is delivered as a .zip file. The SCORM Content Aggregation Model (CAM) book, where content packaging is described, does not define how your SCO files are structured within the content package. The CAM book only requires that the manifest file appear at the top level of your content package folder.

After the package is assembled, you should test the package to ensure it functions the way you had intended. Refer to *Section 9–Testing Your SCORM Content* for more information on testing.

9   Testing Your SCORM Content

You will learn about the difference between compliance and certification and get an overview of the SCORM Test Suite (TS) and Sample Run-Time Environment (SRTE) in this section. ADL has developed tools to help ensure your content is interoperable. These tools also help you examine how your learners experience the content.

9.1   Quality Review

Before using the ADL tools to verify the interoperability of your content, you should conduct a thorough quality review of your content/products to avoid deploying content that contains errors or bugs.

Your internal quality checks should ensure that the content:

- Teaches the defined learning objectives.
- Adheres to your organization's style requirements.
- Is free of spelling and grammatical errors.
- Renders each page as intended.
- Adheres to required compliance issues (e.g., Section 508).

You should also verify that the media elements correspond to the instruction and that SCORM data collection is working as defined in the design documents.

> Carefully check the assessment items to ensure that:
>
> - The accurate correct response is set on each item.
> - The learner passes or fails the assessment when a certain threshold is achieved or not achieved.
> - Remediation corresponding to assessment performance is delivered correctly.

9.2   Compliance and Conformance in SCORM

Since SCORM 2004 was published, several specifications in SCORM documentation have been accepted as IEEE standards. When using standards, the appropriate term is compliance. You either comply with the standard, or not. The path to certification is compliance. Conformance is not an option. Accordingly, ADL will be amending and updating publications to reflect SCORM-compliance. The SCORM Conformance Test Suite will be renamed as the SCORM Test Suite. This change will be reflected in a future reissuance of DoDI 1322.26. ADL recognizes that there are many documents within our archives that cite conformance. Most are in .pdf or other file formats that cannot be changed. The *ADL Guidelines for Creating Reusable Content with SCORM 2004* is the first instance of this change in terminology.

9.2.1   SCORM Compliant

"Compliant" indicates a valid implementation of SCORM. It is possible to be compliant without being certified. "Compliance" indicates that instructional materials and learning management systems have been developed in accordance with the guidelines of SCORM. Vendors can use the Test Suite to determine if their products are "compliant."

9.2.2   SCORM Certified Products

"Certification" indicates that materials have been tested by an independent third party in the Test Suite to assess compliance with the guidelines established in SCORM. All "certified" products are "compliant", but not all "compliant" products have been "certified." The figure to the left depicts the ADL Certified Products logo that you can use in or on your product following receipt of the successful certification test log from an ADL Certification Testing Center.

Since *DoDI 1322.26 Development, Management, and Delivery of Distributed Learning* requires either a valid test log or certification for acceptance of content and systems, some LMS/LCMS vendors use the certification process as a way of proving to their customers that the product was tested by an unbiased, authorized testing center.

Certification is not an endorsement from ADL. It means that the XML manifest was valid and SCOs launched and were able to communicate with the LMS. It does not mean that the LMS has all of the functionality your organization requires or that it uses the data collected from your SCOs. Nor does it mean that the LMS is considered "approved," "good," or a "safe purchase." Likewise, it does not mean that certified content will work the way you want it to in the LMS or that it will run on any other platform, configuration, etc. Each ADL Certification Testing Center determines the fees associated with the certification process.

> A list of certified products may be found at:
> http://adlnet.gov/scorm/certified/index.aspx

9.3   Using the ADL SCORM Test Suite (TS) to Verify Compliance

The TS allows you to self-test LMSs, SCOs, and content packages to determine if they are SCORM compliant. By self-testing your SCOs and content packages, you can help to ensure that the product you deliver will work in your system as intended. The TS is designed to test the requirements outlined in the Requirements document associated with SCORM. It systematically tests the requirements and produces a report that indicates whether or not the product complies with the SCORM requirements. The report has several indicators that will help you understand the results. Table 9.1 lists and describes these indicators.

Table 9.1 ADL SCORM Test Suite (TS) Indicators

Indicator	Issue	Text Color	Sample Message
⊘	**Test Failed Message** This message indicates a violation of a requirement that must be addressed immediately.	Red	ERROR: The SetValue() method call failed
STOP	**TS Termination Due to Non-Compliance or Error Message**	Red	TS Termination due to non-compliance
⚠	**Warning Message** This is an informational warning that something should be repaired or corrected. They are not compliance violations. However, they should be analyzed. Orange may indicate that a recommended SCORM best practice was not followed.	Orange	WARNING: Collection SPM exceeded
✓	**Compliance Check Passed Message** The requirement was met. You do not need to take any action.	Green	The Initialize() method finished successfully
ℹ	This is an informational message that describes what is being tested and where the product is in the testing process.	Blue	Initialize("") has been called
N/A	This message states the compliance level.	Purple	The SCO is SCORM 2004 complaint, as tested in accordance with the SCORM 2004 Test Suite Version 1.0.2 The SCO is SCO RTE 1.0 compliant

Remember, a successful outcome with the TS does not constitute ADL Certification. You can download the TS at http://www.adlnet.gov/downloads/downloadPage.aspx?id=277.

9.4   Using the ADL SRTE to Test Content Delivery and Data Collection
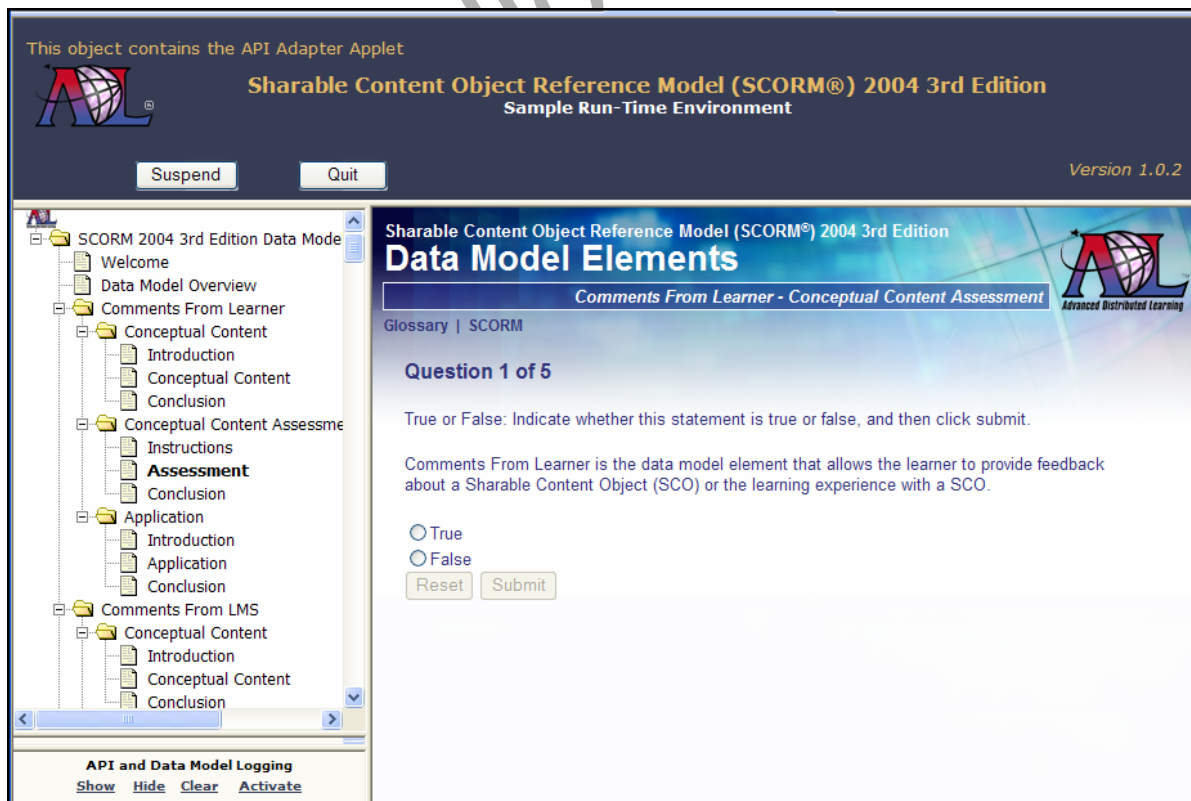
After you have tested your content in the TS, you are ready to test your content in a simulated LMS to see how it will display. The ADL Sample Run-Time Environment (SRTE) is an example of a basic LMS that has implemented the SCORM data model and the minimum functionality specified in SCORM to achieve compliance. You can use the SRTE to display and sequence your content in the same way that learners would experience it. If your content successfully passes using the TS, then it should run properly in the SRTE. Whether you are using the SRTE, or any other SCORM-compliant LMS, you should verify that your content sequences as you intend and that your data model elements are getting and setting the correct values.

SCORM does not mandate all of the functionality that an LMS can provide, in fact, if you tried to build an LMS using only the functionality SCORM specifies, learners would not be able to login, authenticate, etc. SCORM does not address how content should be displayed—in a pane on the current screen, in a new window, full-screen, etc. If you are developing content that will be deployed in a variety of LMSs, then testing your content in the SRTE in addition to testing it in the target LMSs will help you see the different ways your content may be displayed to learners. You can then adapt or adjust your design to ensure you achieve the best experience for learners on the widest variety of systems. Figure 9.1 depicts content running in the ADL Sample Run-Time Environment.

> **Key Point**
>
> If you are developing content that will be deployed in a variety of LMSs, then testing your content in the SRTE in addition to testing it in the target LMSs will help you see the different ways your content may be displayed to learners.

Figure 9.1 Content Running in the ADL Sample Run-Time Environment

The SRTE is not a substitute for testing your content in the TS. You should first test your content in the TS and then in the SRTE.

9.5   Testing in the Target LMS

After you test your content in the TS and the SRTE, load the validated content package onto the LMS you plan to deploy it on. Typically, an LMS will have a testing or "staging" area and a live or "production" area. The staging area can only be accessed by your internal team, not by your learners. Deploying your content in the staging area allows you to run and test the content as if you were an actual learner—to see the content as your learners will see it before giving them access to it. If your content validates in the TS, then it should run properly in the target LMS. Always verify that the content sequences as you intended and the data model elements are getting and setting the correct values and that scores are being rolled-up properly in the target LMS. If your content relies on proprietary functionality from the LMS, such as the use of prerequisites, test that functionality in addition to the content package you are delivering. If you are still unable to access the target environment, then at least ensure that you've tested the content in the SRTE.

Key Point

If your client does not want to give you access to test your content in the target LMS, reiterate the importance of testing the content in the same environment in which learners will experience it.

Refer to http://www.adlnet.gov/colabs/alexandria/index.aspx to contact the ADL Co-Laboratory Hub in Alexandria, Virginia to inquire if the target LMS is available at the Co-Lab for testing purposes at no charge. While it may not exactly replicate the look of the target LMS, the LMS in the Co-Lab will mirror the functionality the target LMS provides.

10 Creating and Registering Content Metadata

You will learn about searching the ADL Registry and writing good metadata to contribute objects to the Registry in this section. The ADL Registry helps users meet two of ADL's architectural functionality requirements: accessibility (the ability to discover and access content) and reusability (taking content designed or developed by others and using it again or in new ways). Good metadata is essential for a search to be effective.

10.1 The ADL Registry

The ADL Registry is the central search point where you can go to discover and acquire DoD training, education, performance, and decision-aiding content that can be redeployed, rearranged, repurposed, and rewritten. In much the same way that a card from the card catalog contains descriptive information about books in a library, the ADL Registry contains all of the registered entries that contain metadata about the content in a repository.

> Metadata is literally data about data. It is the information that describes content, both the individual components (the assets and SCOs) and the content packages.
>
> Information in this section was derived from *The ADL Registry and CORDRA, Volume 2: ADL Registry Overview and User's Guide.*

The ADL Registry provides centrally searchable information, in the form of metadata records (not actual content). The metadata describes many different kinds of objects to enable their discovery and reuse regardless of their location or origin. When you contribute an object to the ADL Registry, you are not "checking-in" the actual object itself. Rather, you are registering the metadata about your the object, along with an identifier that points to the location of the object that you are managing in your local repository.

The ADL Registry provides highly-targeted search results, significantly reducing the number of irrelevant matches because content:

- Is intentionally made visible and searchable by the contributor who is best able to accurately describe it and has vetted it.
- Can be available for search and discovery without providing direct access to it.
- Can be discovered even when it is not accessible to general web search engines.

Generally, content owners want their content to be found. The act of registering content means that the owner wants the content to be discovered and retrieved. The ADL Registry provides a means to discover what contextually appropriate content is available and where it can be found.

Content searchers typically have specific criteria in mind. Searchers often know what they need based on specific requirements. This might be expressed with a simple description, keywords, a specific skill or piece of knowledge, a relationship to other processes, or the state of a learner's profile. The ADL Registry provides a means to relate context-based search criteria to descriptions of specific content, such as mapping a skill definition to an object designed to address that specific skill.

Most content searchers want only what they need. Searchers want the content they find to be as precise a match to their requirements as possible. They do not want every object that **might** pertain; they want the ones that **do** pertain. The ADL Registry provides a means to ensure that discovered content is relevant, accredited, authorized, and deliverable.

With the ADL Registry, you can register documents, assets, SCOs, aggregations, and other useful training or performance-aiding objects. The ADL Registry also encourages the discovery and reuse of those objects by others in DoD and, where permitted, the general public.

Throughout DoD, organizations and agencies operate independently-managed content repositories that have no direct affiliation with one another. These repositories may differ widely in their implementation, management, usage, and access policies. The ADL Registry does not require changes to any of these aspects of locally controlled repositories, but it does require a consistent method for registering metadata in the ADL Registry.

10.1.1 Submitting Metadata Records

The ADL Registry allows DoD Components and their contractors to submit metadata records about the objects they create or acquire. When you submit a metadata record, you are registering the item that the metadata describes. Each object is assigned a unique identifier that serves as a constant reference to the object despite changes in its location across its lifecycle. A single object may have multiple metadata records associated with it through its identifier.

This can occur when multiple parties submit different metadata records for the same uniquely identified object. These metadata records are:

- Expressed using the IEEE Learning Object Metadata (LOM) standard.

- Encoded as Extensible Markup Language (XML™).

- Submitted manually through the ADL Registry web site.

- Submitted automatically through a system developed to interface with the ADL Registry.

10.1.2 Searching the ADL Registry

The ADL Registry web site, as depicted in Figure 10.1, is used to search metadata stored in the ADL Registry. The search results will provide a way for you to resolve to the actual location of an item in a content repository that is associated with the ADL Registry. However, searching the ADL Registry does not guarantee that you will be able to access or retrieve the content you found, because the locally-operated content repository defines its own access and security requirements.

Figure 10.1 ADL Registry Home Page
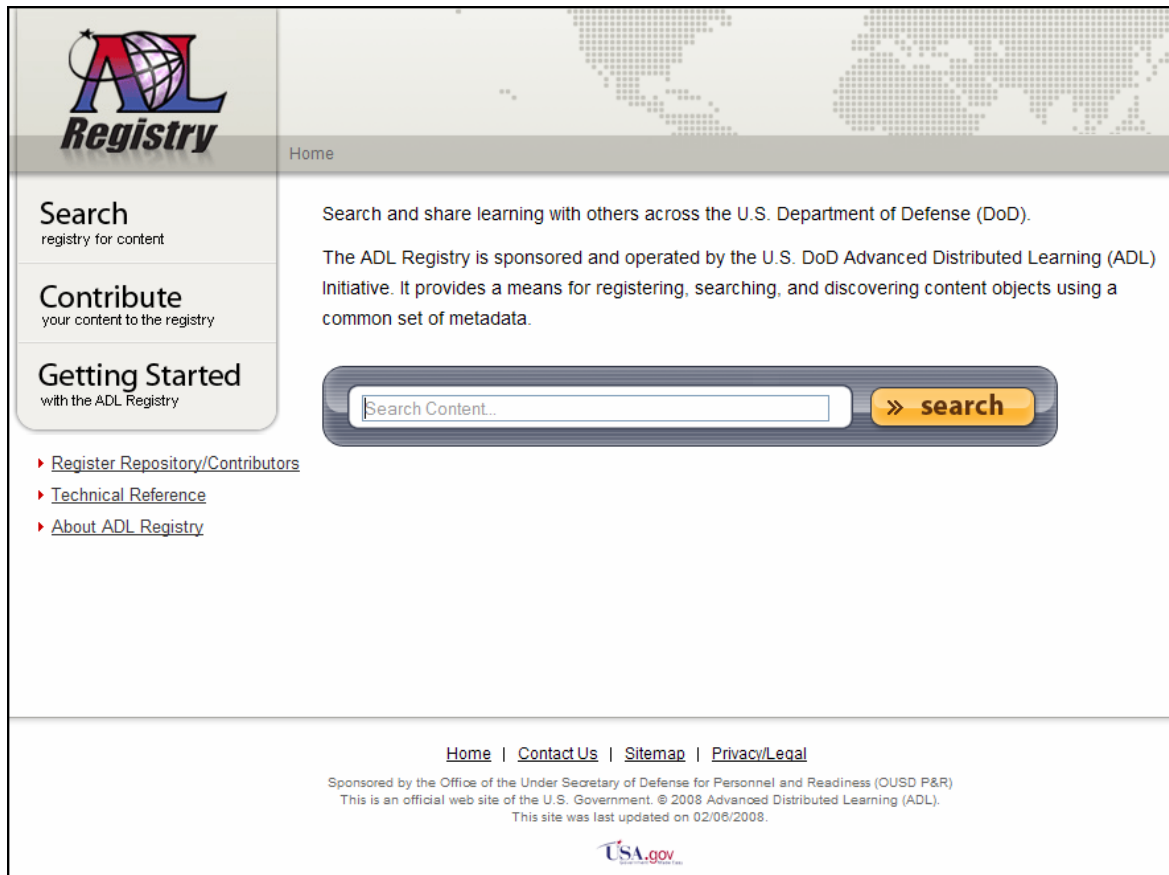
10.1.3 Searching the ADL Registry During Front-End Analysis

DoD Instruction (DoDI) 1322.26, Development, Management, and Delivery of Distributed Learning requires content authors, as part of a Front-End Analysis (FEA), to search for existing content in the ADL Registry before creating new content. The goal of *DODI 1322.26* is to make quality learning content more widely accessible, and to reduce costs by allowing the redeployment, rearranging, repurposing, and rewriting of content instead of constantly developing new content. Refer to *Section 4–Incorporating SCORM into Your Process*, Section 4.2.1 for more information on conducting an FEA.

10.2 Contributing Content to the ADL Registry

To contribute content to the ADL Registry, you must have an account. To obtain an account, go to http://adlregistry.adlnet.gov/register/. One requirement of *DoDI 1322.26* is for content authors who acquire or develop SCORM-compliant content packages, to create metadata, store the packages in a content repository, and register them in the ADL Registry. However, that requirement does not mean that only SCORM content packages can be registered in the ADL Registry.

10.2.1 Objects to Register

In addition to what *DoDI 1322.26* requires, you should register objects that others could redeploy, rearrange, repurpose, and rewrite. The scope of what you register may vary, it may equate to a course, a module, a unit, a lesson, a topic, etc. Sometimes you will register a single SCO in a content package, while other times you may register a content package that contains several SCOs or numerous SCOs with a complex sequencing structure. You may also want to register a simulation, video, animation, or image that would be applicable to, and could be used by, other organizations. You don't need to register every piece of media you create, simply register those that are likely to be reused. Some topics will naturally lend themselves to being redeployed, rearranged, repurposed, and rewritten, so consider registering objects in the areas described in Table 10.1.

Table 10.1 Suggested Items to Register in the ADL Registry

Description	Example
Content that is applicable to, or required by, multiple services or other organizations within DoD	Combating Trafficking in Persons
	Combat Care for Blast Injuries
	Tactical: Patrolling, Defensive Operations
Content that is not DoD or government specific	Accounting
	Leadership
	Medical/Dental/Pharmaceutical
	Regulatory (compliance topics like job safety and sexual harassment)
Content that is governed by Federal policy (instead of just DoD policy)	Transportation of Hazardous Materials
	Federal Acquisition Regulations
	Air Traffic Control
Any content that could be useful to others	Not weapon-system specific
	Principles of Navigation (flight or naval training)
	HMMVEE Maintenance
Any guidance that might be useful to others and that could be packaged in an interchangeable way (such as a PDF file)	Style Guides
	Glossaries
	Reference Manuals

10.2.2 Local Metadata Requirements

While the ADL Registry provides a standard metadata schema based on the IEEE LOM, your local metadata requirements may vary. At a minimum, to register metadata in the ADL Registry, you must use all of the fields it requires. You may be required to produce additional metadata by your organization. If that metadata is LOM compliant, then it will be accessible via the ADL Registry.

10.3 ADL Registry Metadata

The mandatory metadata that is required by the ADL Registry to contribute an item is a subset of the IEEE LOM standard and has been adapted for use in the ADL community of practice. You can use additional LOM elements to more fully describe your objects since the ADL Registry indexes all metadata so that others can find, redeploy, rearrange, repurpose, or rewrite your content.

The descriptions and keywords you provide are essential for effective searches. Use the most specific and descriptive terms possible for the potential audiences because different groups may describe the same content in different ways. For example, some car manufacturers refer to the window-like option located in the roof of the vehicle as a sunroof, others refer to it as a moonroof or vista-roof. In order for individuals searching for vehicle options to find related training content, you would need to use as many of the relevant terms as possible in the keyword and/or description fields.

Another option for ensuring that the right people find the right content with your metadata is to generate multiple metadata instances for a given piece of training content. A metadata "instance" is a single, unique record that references a piece of content. Using the sunroof example above, you would generate a metadata record targeted at people searching for a sunroof, another metadata record for people searching for a moonroof, and another for people searching for a vista-roof. The ADL Registry uses the object identifier (which is the same for each metadata instance) to relate each entry to the other metadata records for the same object.

A metadata record should only reference one specific object. If you create multiple versions or revisions of an object, each iteration must have its own unique metadata. You can accomplish this by assigning each version an identifier and updating the version field in the metadata for every version.

For example, Figure 10.2 depicts an actual metadata record, in human readable format, as found in the ADL Registry.

Figure 10.2 Actual ADL Registry Metadata Record

View Metadata

[download XML file]

Content Information

Title:	**Medical Aspects of Blast Injuries**
Description:	**This course is part of the University of Pittsburgh Supercourse. The course was created by Matthew D. Sztajnkrycer, MD, PhD, Assistant Professor of Emergency Medicine at the Mayo Clinic and Amado Alejandro Baez MD Msc of the Mayo Clinic. The course addresses both the pre-hospital and hospital treatment of blast injuries as well as special scenarios for blast response. The course is registered here with permission from the Supercourse.**
Keywords:	**blast injuries** **IED injury treatment**
Version Number:	**1.0**
Content Status:	**development or acquisition completed**
Collection:	**DOD**

Contributor Information

Contributor Name:	**Nina Pasini Deibler**
Contributor Role:	**author**

Restriction Information

Copyright Restriction:	**no**
Security Level:	**unclassified**
Distribution Restriction:	**LR**

Technical Information

Content Type:	**aggregation**
Object Contents: (MIME/Types)	**text/html** **image/jpeg**
Metadata Schema:	**LOMv1.0** **ADL-Rv1.0**
Conformance:	**SCORM 2004**

Identifiers

Registry ID:	**100.3/registry**
Repository ID:	**100.51/jadlrepository**
Content Object ID:	**100.50.10.13/supercoursesample22**
Metadata Instance ID:	**100.3/MDsupercoursesample221194967522111**

Table 10.2 lists the metadata elements that are required by the ADL Registry.

Table 10.2 ADL Registry Required Metadata Fields

Metadata Element	Description
Title	The official descriptive title assigned by your organization to the object.
Description	Succinct text description of the resource using words and phrases one would search for.
Keyword	Words or short phrases used to identify or define the object.
Version	The current edition or iteration of the object.
Status	The current status or state of the object.
Contributor Role	The individual or organization that authored the object.
Contributor Entity	Contact information for the individual or organization that authored the object.
Contributor Date	The date related to the specific contribution.
Metadata Schema	The name and version of the XML specification used to create the metadata instance.
Technical Format	The format types used in the object.
Copyright	Identifies copyright restrictions on use of, or access to, the object.
Security	Indicates the security classification of the object.
Content Type	Describes the type of content this object is defined as when registered.
Distribution Restrictions	Identifies any distribution restrictions for the object being registered.
Conformance	Describes the type of content this object is defined as when registered.
Categorization	Describes the DoD categorization of use.

For additional guidance on using the ADL Registry and writing metadata for the ADL Registry, refer to *The ADL Registry and CORDRA, Volume 2: ADL Registry Overview and User's Guide*; available on the ADL Registry web site at http://adlregistry.adlnet.gov/about/developers/.

This page intentionally left blank.

11 Glossary

You will learn about definitions for content described in this Guide.

Accessible

Accessible content can be loaded and accessed when needed to meet training and education requirements.

Adaptable

Adaptable content can be customized for individual learners and organizations as needed.

Advanced Distributed Learning (ADL)

An evolving, outcomes-focused approach to education, training, and performance aiding that blends standards-based distributed learning models emphasizing reusable content objects, content and learning management systems, performance support systems/devices, web applications services, and connectivity.

Advanced Distributed Learning (ADL) Registry

The ADL Registry is the Department of Defense (DOD) central registry for content repositories and SCORM content packages. The ADL Registry portal at http://adlregistry.adlnet.gov/ is where DoD affiliated persons, as instructed by *DoDI 1322.26*, submit and search for SCORM content packages.

Aggregation

Aggregations are used to group related content so that it can be delivered to learners in the manner you prescribe. Sequencing rules allow you to prescribe the behaviors and functionality of the content within the aggregation as well as how the aggregation relates to other SCOs within the same root aggregation.

Application Programming Interface (API)

The SCORM API is a standardized method for a sharable content object (SCO) to communicate with the learning management system (LMS) when a learner is interacting with a SCO. There is a specific set of information the SCO can set or retrieve. For example, it can retrieve information, such as a student name, or a set values, such as a score.

Asset

Assets are electronic representations of media, text, images, sounds, web pages and other pieces of data that can be delivered to a Web client. They do not communicate with the LMS. Assets, like the sharable content objects (SCOs) in which they appear, are highly reusable. In order to be reused, assets are described using metadata so that they are both searchable and discoverable in online content repositories.

Authoring Tool

An authoring tool, in interactive courseware, is software that allows an author to generate an instructional program without any explicit programming simply by specifying the instructional content and teaching logic. It combines the components of storyboards into a structured lesson with defined student interactions displayed via computer. It provides an actual lesson framework with an implicit or explicit teaching strategy.

Certification or Certified

"Certification" indicates that materials have been tested by an independent third party to assess conformance with the guidelines established in SCORM. "Certification" indicates a successful testing by the Conformance Test Suite. All "certified" products are "conformant."

Configuration Management (See also lifecycle management.)

Configuration management is defined as a process for establishing and maintaining consistency of a product's performance, functional and physical attributes with its requirements, design and operational information throughout its life.

Compliance or Compliant

A product is compliant when tested to ensure it performs according to applicable guidelines, instructions, policy, or law. The SCORM test suite is designed to rigorously test inputs, processes, and outputs. Non-compliance (e.g., to a contract) may result in remedial and/or punitive actions. For more information, refer to ADL's position on compliance and conformance in the Using This Guide section of this document.

Conformance or Conformant

A product or service is conformant when it adheres to technical specifications, guidelines, recommendations, or best practices to identify the correctness, completeness, and quality of developed product or service. Test assertions are achieved by inspecting results focused on reliability, stability, portability, maintainability, and usability. No form of testing is used other than evaluating actual results against expected results. For a more complete listing of attributes consult International Organization for Standardization standard ISO 9126. For more information, refer to ADL's position on compliance and conformance in the Using This Guide section of this document.

Content Package

A content package is a standardized way to exchange collections of digital resources between different learning management systems (LMSs), authoring tools, content repositories, and operating systems. In traditional instructional design terms, the content package would be everything needed to deliver the course, module, lesson, etc. to the learner. The content package contains two principal entities: (1) a manifest file that lists all of the resources or assets you want to include in the package, the content structure diagram you created (called the organization), the sequencing rules, and all of the metadata for the SCOs, the LOs, and the package itself; (2) all of the physical SCO and asset files for the content package.

Content Repository

An accessible digital storage system containing SCORM content packages.

Content Structure Diagram

A content structure diagram is a tree diagram created by the instructional designer for the programmer to show the hierarchy onto which the sequencing rules for the SCOs are applied. The diagram should be followed by a list of behaviors the instructional designer intends for the learner.

Content Object Repository Discovery and Registration Architecture (CORDRA)

CORDRA is an open, standards-based model for the design and implementation of software systems for the purposes of discovery, sharing, and reuse of learning content through the establishment of interoperable federations of learning content repositories. It is used to provide access to high-quality educational resources, training performance aids, and pedagogies that may be shared and tailored to meet individual learner needs. The ADL Registry is the first publicly available CORDRA registry.

Data Model Elements

A set of information about a learner's performance in, and interaction with, the instructional content initiated by the SCO and stored in an LMS. Data model elements are made interoperable by the SCORM Run Time Environment Data Model. This data is the only information you are able to collect in an interoperable manner.

Distractor

In multiple choice items, a series of response options follows the stem. One of the response options is the correct answer; the others are called distractors.

Durable

Durable content does not require modification to operate as versions of software systems and platforms are changed or upgraded.

Instructional Strategy

An instructional strategy consists of the learning methods employed in the content, the assets, and media used in and for delivery of the content and any testing, interaction, or structural strategies (such as remediation) that are used to help learners attain or achieve the desired instructional outcomes.

Interoperable

Interoperable content operates across a wide variety of hardware, software, operating systems, and web browsers regardless of the tools used to create it and the platform on which it is initially delivered.

Item

Test questions are commonly called items, because the term "item" includes test questions that are not strictly questions in the grammatical sense. For example, a completion statement, while not in the form of a question, is a valid type of test item.

Learning Content Management System (LCMS)

An LCMS is a development of the learning management system, it is a multi-user environment where learning developers may create, store, reuse, manage, and deliver digital learning content from a central object repository.

Learning Management System (LMS)

An LMS is a software package used to administer one or more courses to one or more learners. An LMS is typically a web-based system that allows learners to authenticate themselves, register for courses, complete courses and take assessments. The LMS stores the learner's performance records and can provide assessment information to instructors.

Learning Object

Any entity, digital or non-digital, that can be used, re-used, or referenced during technology supported learning.

Manifest

A manifest is a description of everything contained in your content package. Generally a tool such as the Reload Editor, will create the manifest as an XML document during the content packaging process. The manifest includes all of the resources or assets included in the content package, the content structure diagram you created (called the organization), the sequencing rules, and all the metadata for the SCOs, LOs, and the package itself.

Metadata

Metadata is "data about data." It is the information that describes what your content is, both the individual pieces (the assets, SCOs, LOs) and the content packages. Metadata enables instructional designers searching for content or assets to locate them with relative ease and determine whether they will be useful before downloading or requesting rights to your content. The ADL Registry has a custom metadata taxonomy to which all metadata registered in it must adhere.

Navigation

Navigation is the means by which learners move through multimedia content.

Objective (learning)

A learning objective is a statement that describes the content to be taught, and the learner behavior and/or criterion of mastery that will validate achievement of the desired learning.

Objective (sequencing)

For SCORM sequencing purposes, an objective is a global variable that allows the learning management system (LMS) to share status values between sharable content objects (SCOs). This gives designers greater flexibility in structuring the content under SCORM guidelines. Depending on the designer's requirements for the instruction, the objective may or may not track actual learner objectives, knowledge, skills, or abilities.

Organization

The organization is the part of a content package where SCOs are ordered into a tree structure and sequencing behaviors are assigned to them. The organization outlines the entire structure you have created for your content. The organization provides order to the otherwise unordered collection of SCOs and their metadata.

Practical Exercise

A practical exercise is an ungraded, unscored test item located within a section of learning content. Learners are allowed two attempts and are provided with immediate feedback on their performance.

Remediation

Remediation is used to help learners comprehend instruction with which they may be struggling. In the event that learners do not answer a test item correctly, or fail an entire test, they can remediate to already viewed content for review or completely new content to try and understand from a different approach.

Repository

A repository is a device for storing and maintaining digital information (content). Content proponents declare the existence of the data chunks in a registry for discovery and retrieval by others.

Repurpose

Repurposed content is existing content used as a baseline to create new content in new or different contexts.

Response Option

In multiple choice test items, a series of response options follows the stem. One of the response options is the correct answer; the others are called distractors.

Reuse

Reused content is existing content used in new or different contexts or applications.

Root Aggregation (See also organization.)

A root aggregation is a top-level aggregation.

Shareable Content Object Reference Model (SCORM)

SCORM is a model that references and integrates a set of interrelated technical standards, specifications, and guidelines designed to meet high-level requirements for e-learning content and systems

Sequencing

Sequencing is similar to the CBT term "branching" in that it describes and prescribes the manner in which learners receive content. In CBT, much of the branching or sequencing occurred within a lesson or course as the learner completed different tasks. However, in SCORM, the learning management system (LMS) sequences all activities between the sharable content objects (SCOs) and the learner, essentially performing all of the sequencing of the content based upon rules created by the designer.

Sharable Content Object (SCO)

In general terms, a SCO is a collection of assets that becomes an independent, defined piece of instructional material. SCOs are the smallest logical unit of instruction you can deliver and track via a learning management system (LMS).

Share

Shared content is existing content used as it was designed and developed without modification.

Stem

The stem is the question or statement at the beginning of a multiple choice test item.

Style Guide

A style guide is the set of established criteria, processes, and procedures a team follows throughout the process of creating instructional materials. It should serve as the handbook or primary reference material for most questions concerning design, layout and standardization that arise during the content development process. A Style Guide should be minimally impacted by SCORM-specific conventions.

Templates (authoring)

Authoring templates are standardized layouts for the authoring of content pages.

Templates (sequencing)

Sequencing templates are pre-programmed instructional strategies that use a standardized set of sequencing rules. The sequencing templates described in this Guide were developed by the Learning Systems Architecture Lab (LSAL) at Carnegie Mellon and were programmed into the Reload Editor by ADL staff.

This page intentionally left blank.

12 References

You will learn about reference resources that contributed to the content in this Guide.

ADL Registry and CORDRA, Volume 2: ADL Registry Overview and User's Guide. (2008, Department of Defense).

Blackmon, William H. and Roberts, Eric J. SCO Sighs: Why ADL Won't Say How Big SCOs Should Be. (2006, Interservice/Industry Training, Simulation, and Education Conference (I/ITESEC) Paper Number 2943).

DoD Instruction 1322.26, "Development, Management, and Delivery of Distributed Learning." (June 16, 2006).

DoD Instruction 1322.20, "Development and Management of Interactive Courseware (ICW) for Military Training." (March 14.1991).

*Joint Advanced Distributed Learning Co-Laboratory Terms and Definitions
http://www.jointadlcolab.org/help/terms/index.aspx (2008).*

Learning Systems Architecture Lab (LSAL). Defense Ammunition Center e-Learning Style Guide 1st Edition. (2005, Carnegie Mellon University).

Learning Systems Architecture Lab (LSAL). SCORM Best Practices Guide for Content Developers 1st Edition. (2003, Carnegie Mellon University).

Microsoft Corporation. Microsoft Manual of Style for Technical Publications 3rd Edition. (2004, Microsoft Press).

Military Performance Specification 29612B, "Training Data Products." (August 31, 2001).

This page intentionally left blank.

13 Authors and Contributors

You will learn about the authors and contributors of this Guide.

Nina Deibler, SI International, Inc.

Nina is a Principal Consultant in the Learning Solutions division of SI International, Inc., a provider of information technology and network solutions primarily to the Federal government. She has over 12 years of experience designing and developing learning solutions for clients in academia, industry, and government. From 2000 to 2006, Nina was part of the Learning Systems Architecture Lab (LSAL) at Carnegie Mellon University where she co-authored the several SCORM related publications including the "SCORM Best Practices Guide for Content Developers". Nina also designed and co-led a successful series of workshops to assist content developers and programmers implementing SCORM-conformant instructional materials. Nina's other experiences include proposal writing, consulting, and project management. Nina served as a senior instructional designer on the Boeing C-17 Aircrew Training System designing and developing computer, instructor, and simulator-based training for pilots and loadmasters for four years. Nina earned an MA from Abilene Christian University and an MAS from Embry-Riddle Aeronautical University. She received her BA from Grove City College. Nina is currently evaluating and designing instruction and supporting the knowledge management efforts of the U.S. Army Defense Ammunition Center (DAC). She also supports the outreach and technical efforts of the ADL Initiative.

Paula Dosch-Haworth, Booz Allen Hamilton

Paula is a Senior Consultant and Instructional Technologist with Booz Allen's Organization and Strategy Human Capital Learning Team, a leading global consulting firm that helps government clients solve their toughest problems with services in strategy, operations, organization and change, and information technology. She has 12 years of experience designing and developing learning solutions for government, industry, and not-for-profit clients. Paula has designed learning programs for an array of government clients in the civilian and defense sectors. She taught Navy doctors to treat biological warfare cases on the battlefield during the anthrax terrorist attacks, and trained Officers who work in international trade for Customs and Border Protection (CBP). Prior to her work in Washington, Paula worked for an international communications firm in Japan, supporting government agencies and private corporations entering or expanding operations in the Japanese market. Paula holds a BA from the University of Michigan, an MA in Instructional Technology from the George Washington University, and recently earned the credential of Certified Professional in Learning and Performance (CPLP) from the American Society for Training and Development (ASTD). Paula is currently also working with the Smithsonian Institution's National Museum of Natural History (NMNH) to design and develop an innovative Webcomic mystery story that complements a Smithsonian exhibition "Written in Bone" on forensic anthropology in the Colonial Chesapeake region.

Aimee Norwood, Booz Allen Hamilton

Aimee is an Associate with Booz Allen Hamilton's Organization and Strategy Human Capital Learning Team. She has over 15 years of experience in learning project management, developing and delivering solutions for government and industry clients. From 2003 to 2005, Aimee was part of the organic ADL Initiative Team leading Outreach efforts where she contributed to ADL and SCORM publications and PlugFest events. Aimee helped to establish the Training Transformation (T2) Joint Knowledge Development and Distribution Capability (JKDDC) Knowledge Development Branch as the Chief during it's initial operating year. During this time, she was responsible for DoD individual joint education and training content development and evaluation. Aimee received her BA in Human Development from Mount Vernon College. In addition to client consulting work, Aimee is a lead on Booz Allen's Immersive Learning Service Offering that focuses on improving human performance through learning, technology, and supporting strategies.

Krystal Spitz, Booz Allen Hamilton

Krystal is a Consultant with Booz Allen's Organization and Strategy Human Capital Learning Team. She has helped commercial and government clients with e-learning and Instructor Led Training (ILT) interventions ranging from designing learning, producing content development guidelines, and delivering learning via web portals. She is skilled in technical writing, learning development, quality assurance, facilitation, and project management.

Krystal received her BA in English from Mary Washington College. Krystal is currently supporting the Director, Readiness and Training, Policy and Programs, Office of the Under Secretary of Defense for Personnel and Readiness in assessing the rationale for and level of effort provided to Combatant Commanders (COCOMs) by distributed support personnel in order to improve the efficiency and effectiveness of current approaches for allocating these distributed support personnel.

Contributors

We would like to extend our sincere gratitude to the following contributors:

Tom Archibald
Peter Berking
Dr. William Blackmon
Ellen Epstein
G.A. Redding
Dr. Eric Roberts
Kiril Tchangov

Michael Byars
Kathryn Cecil
Mona Sopakar
Victoria Veluz

John Campbell
Kim Payne