S1000D-SCORM Bridge Toolkit

DRAFT
Version 1
23 February 2011

1. Introduction 4
1.1. About this document 4
1.2. The Bridge Project Motivation 4
1.3. S1000D-SCORM Bridge Toolkit 5
1.4. S1000D-SCORM Bridge Toolkit Scope 5
1.4.1. S1000D Support 5
1.4.2. SCORM Support 6
1.5. About S1000D 6
1.6. About SCORM 7
1.7. Toolkit Users 7
1.8. Licensing 7
2. Key Concepts 8
2.1. S1000D-SCORM Bridge Toolkit Overview 8
2.2. Toolkit Building Blocks 8
2.2.1. Common Source DataBase (CSDB) 8
2.2.2. SCORM Content Package Module (SCPM) 8
2.2.3. Data Module (DM) 9
2.2.4. Input Package and Output Package 9
2.2.5. IMS Manifest 10
2.2.6. Sharable Content Object (SCO) 10
3. Getting Started 11
3.1. System Requirements 11
3.2. Setup and Installation 11
3.2.1. Toolkit Distributions 11
3.2.2. Installation 11
3.2.3. Installing the Java JDK 11
3.2.4. Installing the Apache Ant 12
3.2.5. Setting environment variables on Windows 12
3.2.6. Contents of the Toolkit 12
3.2.7. Configure the Toolkit 14
3.3. Try the Toolkit 14
3.4. How Developers Will Use the Toolkit 17
3.4.1. Publishing SCORM 2004 from an S1000D Compliant Application 17
3.4.2. Importing S1000D Content into a Learning Management System 18
3.4.3. Independent Application Development 18
4. Using the Toolkit 19
4.1. The Input Package 19
4.2. Toolkit Local files 19
4.2.1. The ViewerApplication Directory 19
4.2.2. The xsd Directory 20
4.3. Running the Toolkit, Command Line 21
4.4. Transformation 22
4.5. Reviewing the Output Package 22
5. Integrating the Toolkit 24
5.1. Conceptual Architecture Modules 24

5.2. Apache Commons Chain 24
5.3. PRE PROCESS MODULE 25
5.4. SCO BUILDER 27
5.5. POST PROCESS MODULE 28
5.6. VIEWER MODULE 28
6. Troubleshooting 29
6.1. Log files 29
6.2. Error messages 29
6.3. Debugging 29
6.4. Sample files description 29
7. Terms and Definitions 30
8. Points of Contact 34

DRAFT

1. Introduction

1.1. About this document

This documentation covers conceptual, usage, integration and troubleshooting information needed to operate the S1000D-SCORM Bridge Toolkit, an open source tool that converts S1000D 4.0 data into SCORM 2004 content packages. The Toolkit is one deliverable from "The Bridge Project." The Bridge Project's objective is to reduce data ownership costs and increase production efficiencies by integrating technical data and training content production in a common source database (CSDB). The Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD(AT&L)) funded the project from its Reduction in Total Ownership Cost (RTOC) program. Advanced Distributed Learning (ADL) Initiative, Office of the Secretary of Defense for Personnel and Readiness (OSD(P&R)) managed the Bridge Project.

1.2. The Bridge Project Motivation

The goal to educate and train in rapidly changing technical environments requires information products of many types to be efficiently updated by and distributed to the right people at the right time and in the right format. To reach this goal, organizational practices that have historically delayed training content production until after products and systems are deployed must share common business processes and infrastructure, such as data acquisition, production, management and delivery. Sharing business practices and infrastructure is now possible with the release of the S1000D Technical Data Specification, Issue 4.0. The release provides a data strategy that allows learning data to be expressed in a neutral way that directly ties content to products, components, and doctrines. S1000D is an ideal part of a data readiness solution because:

- S1000D has been widely adopted across programs and countries.
- S1000D uses XML, the very nature of which is neutral, nonproprietary and portable.
- S1000D supports data interchange between programs and vendors.
- S1000D ties each item of data to a system component via a Standard Numbering System (SNS), a set of unique configuration codes.

Currently developers create and maintain technical and training content in applications and databases that are not integrated. This lack of integration negatively affects life cycle costs and readiness in these ways:

- Training development timelines can be long due to developmental and operational system testing schedules. Technical data and training content are both dependent on system testing finalization. Direct and timely data reuse and repurposing before, during and after testing is prevented by unintegrated authoring environments.

- Training content is not traceable to authoritative technical content. Thus a lag time exists between the time engineering change specifications are received and training content is identified and updated.
- Training content is often needlessly duplicated due to a lack of authoritative source and system traceability.
- The costs of development and update of training are not factored into life cycle costs due to lack of standard processes tied to authoritative technical content.

The Bridge Project will provide a collection of tools that will "bridge" the technical data and learning content domains using S1000D as shared data specification. The Bridge tools will:

- Integrate any learning content authoring tool with any authoritative source database.
- Provide an ability to rapidly adjust learning content in response to changing system requirements.
- Optimize the development process for content reuse.
- Facilitate the integration of technical learning content into life cycle support.
- Transform S1000D learning and technical data modules into SCORM 2004 content packages.

1.3. S1000D-SCORM Bridge Toolkit

The Toolkit is an open source framework and a reference implementation supporting the transformation, packaging and viewing of S1000D data into SCORM 2004 3rd Edition content packages. The functionality offers a consistent method for systems-based training developers to create technical learning content using traceable S1000D asset formats grouped into learning modules and transform the assets and grouped learning modules into organized SCORM 2004 packages. S1000D-based XML aggregation structures are mapped into the IMS manifest. S1000D data modules are mapped into HTML.

1.4. S1000D-SCORM Bridge Toolkit Scope

1.4.1. S1000D Support

The Toolkit supports content authored, published and maintained according to S1000D Release 4.0. Current capability has focused on providing full support for learning data modules and the SCPM. Consequently, the viewing stylesheets provided in the Toolkit have only specific reference capabilities for the Descriptive and Procedural data modules in the Bike Sample data. Implementers can extend the framework to allow support of any data module type, by providing their own stylesheets.

Schemas that are out of scope of the initial Toolkit release include:

- Crew/Operator
- Fault

- Parts Data
- Maintenance Schedule
- Process Module
- Container
- Wiring Data
- BREX
- Technical Information Repository
- Checklists

NOTE: The Toolkit does not provide a comprehensive S1000D content viewer. The viewer application provided with the Toolkit is designed to specifically to support the S1000D markup found in the S1000DBike Learning Sample. If output content processed by the Toolkit does not render as expected, it may be necessary to customize the viewer rendering transformation files to achieve the desired rendering results.

1.4.2. SCORM Support

Only support for SCORM 2004 3rd Edition conformant output is provided.

The current Toolkit release does not transform S1000D sequencing and navigation features into the SCORM content package. If sequencing and navigation behaviors are applied to the SCORM 2004 3rd edition output, they will not be present in subsequent transformed versions of the same content. These open source features may be added as needed by toolkit users and supported in future releases of the toolkit.

1.5. About S1000D

S1000D is an international specification for the procurement and production of technical publications. It is an SGML/XML standard for preparing, managing, and exchanging equipment maintenance and operations information. It was initially developed by the AeroSpace and Defence Industries Association of Europe (ASD) for use with military aircraft. The standard has since been modified for use with land, sea, and commercial equipment. S1000D is copyrighted by the S1000D Council, which includes board members from ASD, the United States' Aerospace Industries Association (AIA), and the Air Transport Association (ATA). It is maintained by the S1000D steering committee, which includes national industry and defense representatives from most of the countries currently using the standard. See also:

See also:

- http://www.s1000d.org/

1.6. About SCORM

The Sharable Content Object Reference Model (SCORM®) integrates a set of related technical standards, specifications, and guidelines designed to meet SCORM's high-level requirements—accessible, interoperable, durable, and reusable 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.

See also:

- http://www.adlnet.gov/Technologies/scorm
- http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/Previous%20Versions/index.aspx

1.7. Toolkit Users

The primary Toolkit implementer and user communities will be CSDB vendors, LMS vendors, content development tool vendors and application developers. The CSDB vendors can integrate the Toolkit into their CSDB application to output SCORM 2004 conformant training. LMS vendors can integrate the Toolkit into LMS applications to fetch and import S1000D training content from a CSDB. Content development tool vendors can incorporate the Toolkit for developmental testing purposes. Independent vendors will be able to create applications that fetch content from any CSDB and create SCORM 2004 content packages that can be consumed by any SCORM conformant LMS.

1.8. Licensing

The Toolkit is licensed for use, at the user's election, under the Eclipse Public License v 1.0 or the Apache Software Foundation License v2.0.

If, at the time of use, the Project Management Committee, consisting of the Source Forge project team members, has designated another version of these licenses or another license as being applicable to the Toolkit, user may select to have its subsequent use of the Toolkit governed by such other designated license.

A copy of the Eclipse Public License v 1.0 is available at
http://opensource.org/licenses/eclipse-1.0.php

A copy of the Apache Software Foundation License 2.0 is available at
http://opensource.org/licenses/apache2.0.php

This statement found at this URL must be included in any copies of Toolkit code:
http://s1000d-scorm.svn.sourceforge.net/viewvc/s1000d-scorm/license.txt?revision=11&view=markup

2. Key Concepts

2.1. S1000D-SCORM Bridge Toolkit Overview

Conceptually, the Toolkit is a black box that consumes S1000D input files and graphics from a Common Source Database (CSDB) and produces a SCORM 2004 3rd Edition Content Package. The input files consist of a SCORM Content Package Module (SCPM) and referenced S1000D data modules. The Toolkit transforms the SCPM input file into a SCORM 2004 IMS Manifest and creates SCOs from the S1000D data modules. These outputs are packaged into the SCORM 2004 Content Package.

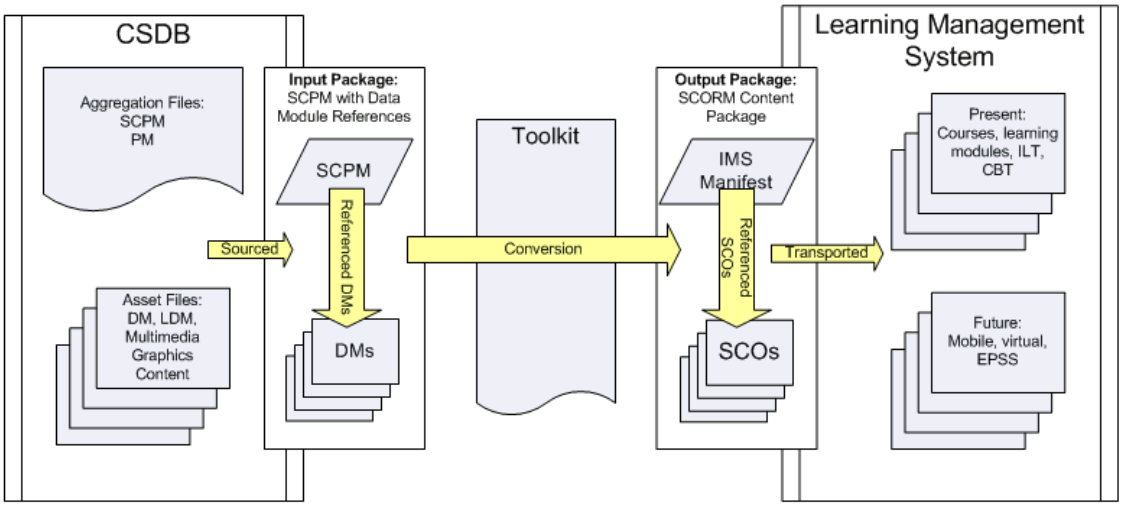

Figure 2.1: Conceptual Model of Toolkit Usage

2.2. Toolkit Building Blocks

2.2.1. Common Source DataBase (CSDB)

The CSDB is a collection of data modules documenting systems and components. According to the S1000D specification, the CSDB is an information store and management tool for all objects required to produce the technical publications and courseware within projects. The CSDB is the source for the inputs to the Toolkit.

See also:

- http://www.s1000d.org/

2.2.2. SCORM Content Package Module (SCPM)

The SCORM Content Package Module (SCPM) is an aggregation file that references S1000D data modules into authoritative course and SCO structures. The S1000D version 4.0 specification defines the SCPM in Chapter 3.2 and Chapter 4.9.5.

The developer organizes the course in the SCPM by referencing S1000D data modules into collections. Each collection is the basis for a SCO. The SCPM and the referenced data modules are the inputs to the Toolkit.

See also:

- http://www.s1000d.org/

2.2.3. Data Module (DM)

Data Modules are configurable, reusable pieces of technical information stored in a CSDB. They are uniquely identified by a Data Module Code (DMC). Each data module contains a metadata section called the Identification and Status section. It identifies and describes the data module. A separate Content section contains the technical content. The S1000D version 4.0 specification defines the Data Module in Chapter 3.2.

The developer can organize the course in the SCPM by referencing data modules into the collection.

See also:

- http://www.s1000d.org/

2.2.4. Input Package and Output Package

The Toolkit input package contains the SCPM and referenced S1000D data modules sourced from the CSDB. The Toolkit output package is a SCORM 2004 3rd Edition conformant zip file. This SCORM 2004 Content Package contains an XML file called an IMS Manifest file, content called SCOs and any required schemas. Section 5 describes the content package in more detail.

The input package represents the authoritative source to the output package. To create the IMS Manifest, XSLT transforms the input package to the output package by mapping the SCPM structures to IMS Manifest structures. To create the physical SCOs, the Toolkit consumes the referenced S1000D data modules and creates them according to the authoritative SCPM organization.

The output package is the SCORM Content Package. It can transport the training content to any Learning Management System that is certified to be compatible with the SCORM 2004 standard.

In the Learning Management System environment, learners can launch and play each SCO, interact with its content and take assessments. The Learning Management System can track overall user course and SCO statuses and provide management reports.

2.2.5. IMS Manifest

The IMS Manifest is an XML document that describes the content structure and associated resources of the SCORM 2004 package. The IMS Manifest contains the transformed elements of the SCPM in a format that an LMS can use.

The IMS Manifest and SCPM have similarities and differences and should be distinguished. The SCPM is an S1000D document within the CSDB that identifies the S1000D data modules, metadata and organizational structure that will be used in the SCORM course. While creation of the SCPM is a needed first step to repurpose technical content into SCORM conformant content, it is not an object that a SCORM course can use or one that a Learning Management System (LMS) can understand or process. The LMS requires the IMS Manifest to describe all the metadata, organizations and resources of the SCORM course. It is agnostic of the source of the content. The Toolkit bridges the two documents by transforming the authoritative organizational data of the SCPM to the IMS Manifest data structure.

See also:

- http://www.adlnet.gov/

2.2.6. Sharable Content Object (SCO)

The IMS Manifest within a SCORM course references and organizes SCOs in the course. Each SCO is a collection of reusable assets that comprise a learning event. It is the lowest level of granularity tracked by a learning management system. Each SCO contains a single launchable learning resource. A Learning Management System that contains a SCORM conformant Run Time Environment can launch each SCO and communicate with it through standardized protocols defined by SCORM 2004 3rd Edition.

When the Toolkit creates SCOs it uses the S1000D data modules referenced by the SCPM for content. The launchable learning resource is an HTML frameset that is used to display the navigation and content. When the Toolkit-produced SCO is viewed in the browser, each document is displayed as one screen. The order of screens depends on the order in which they were originally listed in the SCPM.

See also:

- http://www.adlnet.gov/

3. Getting Started

3.1. System Requirements

The Toolkit is written in Java and requires Java SE 6 (JRE) to be installed to execute the application and the Java code.

3.2. Setup and Installation

The Toolkit package comes with all required tools except the Java JDK and the binary distribution of Apache Ant.

3.2.1. Toolkit Distributions

The Toolkit is currently available in a full source distribution format.

The full source distribution contains the source and executable code for the Toolkit plus all of the documentation describing how to use to Toolkit. This distribution will allow you to modify Toolkit Java code and look at how the Toolkit works.

The distribution is available for download from https://sourceforge.net/projects/s1000d-scorm/

3.2.2. Installation

To install the full package:

- Download the full package from https://sourceforge.net/projects/s1000d-scorm/
- Unzip the package into the C:\ directory on Windows, or into a home directory on Linux.

Note: You cannot run a Toolkit build until you have installed the Java JDK and Apache Ant.

3.2.3. Installing the Java JDK

Installing the JDK on Windows

1. For the Oracle version of the JDK, enter the URL http://www.oracle.com/technetwork/java/javase/downloads/index.html
2. Locate the "Download JDK" button to download the latest version. (The Toolkit was build using Java SE 6).
3. Select the appropriate platform and language for your system. (Accept the License Agreements and complete all necessary forms on the site)

4. Save and install the .exe file.
5. Set the JAVA_HOME environment variable to location of you installed JDK. Ex. C:\Program Files\Java\jdk1.6.0_21

3.2.4. Installing the Apache Ant

Installing Ant on Windows

1. Enter the URL: http://ant.apache.org/bindownload.cgi.
2. On the Apache Ant Project page, find the heading Current Release of Ant.
3. Select apache-ant-1.8.2-bin.zip [PGP] [SHA1] [MD5]
4. Click Save to unzip the apache-ant-1.8.2-bin.zip [PGP] [SHA1] [MD5] file and save it to your C:\ directory as ant.
5. Add the bin directory to your PATH environment variable.
6. Add the ANT_HOME environment variable set to C:\ant.

3.2.5. Setting environment variables on Windows

1. From the Start Menu, select Start > Settings > Control Panel.
2. Double-click System to open the System Properties window.
3. On the Advanced tab, select environmental variables.
4. Modify each environmental or system variable.

Set the PATH environment variable to include the directory where you installed the Ant bin directory:

1. Find the PATH environment variable in the list. If PATH is not listed, click on New under the System variables section.
2. Type %ANT_HOME%\bin;%JAVA_HOME%\bin;

3.2.6. Contents of the Toolkit

Figure 3.2.6 shows the directories and files of the Toolkit. The table following describes each directory.

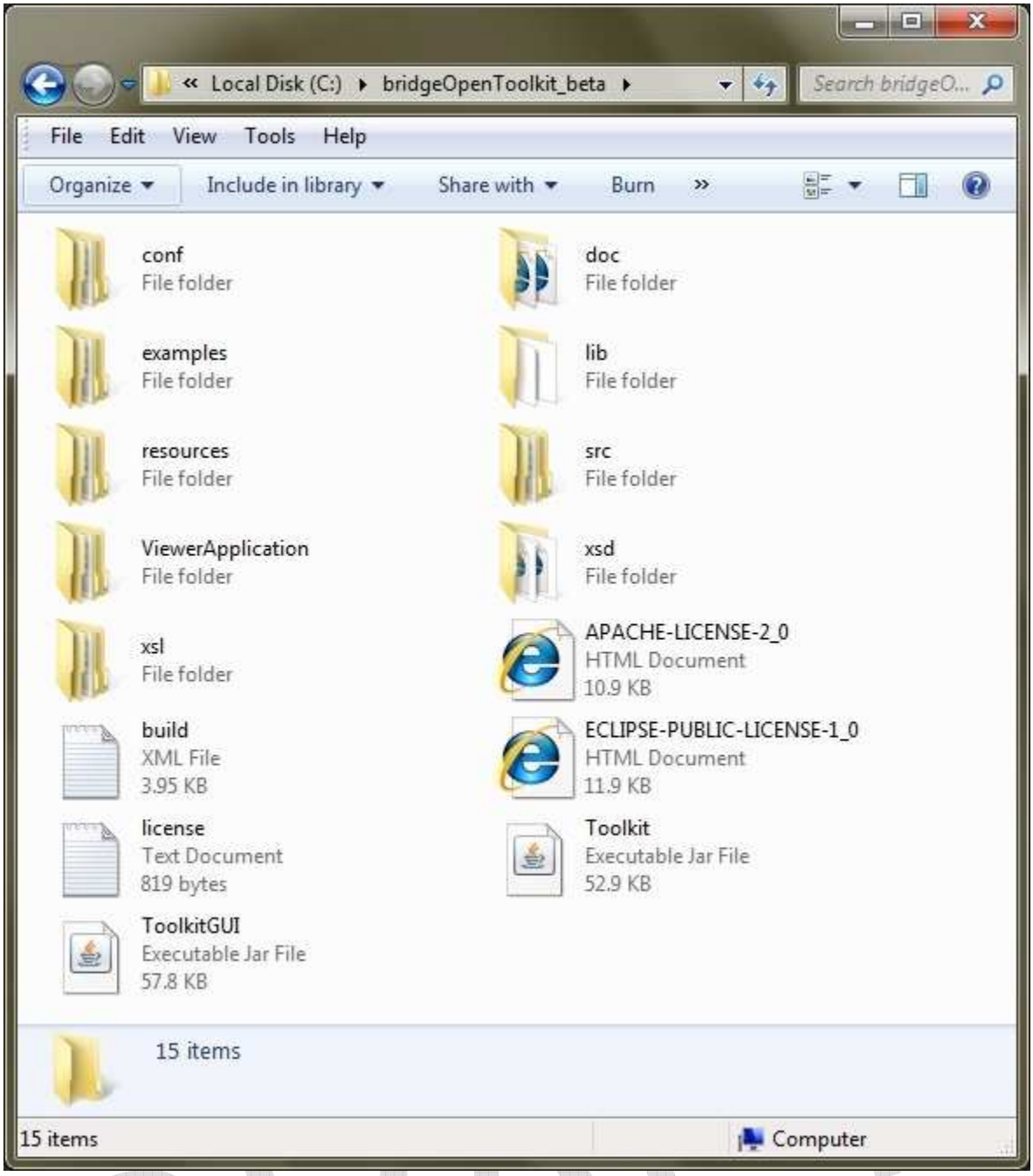

Figure 3.2.6: Toolkit Directories and Files

Directory	Description
root	The root directory contains the Ant build.xml file, license files such as APACHE-LICENSE-2.0.html, ECLIPSE-PUBLIC_LICENSE-1_0.html and license.txt. The executable jar files Toolkit.jar (command line executable) and ToolkitGUI.jar (Java GUI executable).

	This GUI was implemented to give the developer a feel for the functionality of the Toolkit and is not designed to be integrated within a vendor application.
conf	The conf directory contains the Apache Commons Chain Catalog XML document.
doc	The doc directory contains the JavaDoc files.
examples	The example directory contains the S1000D 4.0 example bike data. The bike_resource_package directory contains the data module resource XML files for the Bike demonstration project. It also contains all image files referenced by the resource XML files. The bike_SCPM directory contains the S1000D SCPM for the Bike demonstration project.
lib	The lib directory contains libraries required by the executable. The directory includes jars for the Apache Commons Chain. It also includes jars for parsing, manipulating and outputting XML.
resources	The resources directory contains miscellaneous resource file.
src	The src directory contains the Java source code.
ViewerApplication	The ViewerApplication directory contains the .XSLT, .CSS and other files needed to render the S1000D data modules in a browser.
xsd	The xsd directory contains the ADL and LOM XML Schema files.
xsl	The xsl directory contains the stylesheet that converts the SCPM to an imsmanifest.xml file.

3.2.7. Configure the Toolkit

Instructions to configure the Toolkit will be delivered with the final Toolkit product.

3.3. Try the Toolkit

A Toolkit sample is provided to convert SCPM and resource files for the Bike sample project into a SCORM 2004 Content Package. A GUI interface is provided to demonstrate this functionality. This is not meant to be used in a production environment. Here are the steps to run the GUI and create the output.

Step 1: Locate the jar file, ToolkitGUI.jar, in the Toolkit download. Double-click the ToolkitGUI.jar to reveal the GUI. See **Figure 3.3a.**

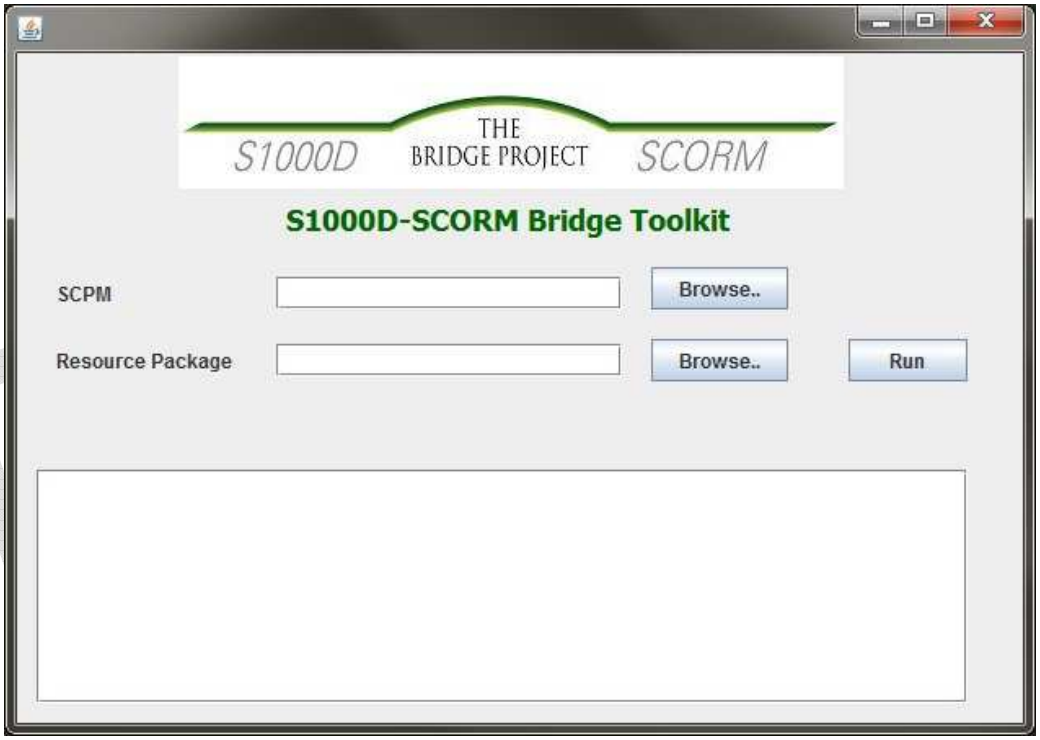

Figure 3.3a: Toolkit GUI

Step 2: Click the browse button to search for the SCPM. Select the SCPM XML file and click Open. The following screenshot shows the sample Bike project SCPM. See **Figure 3.3b**.

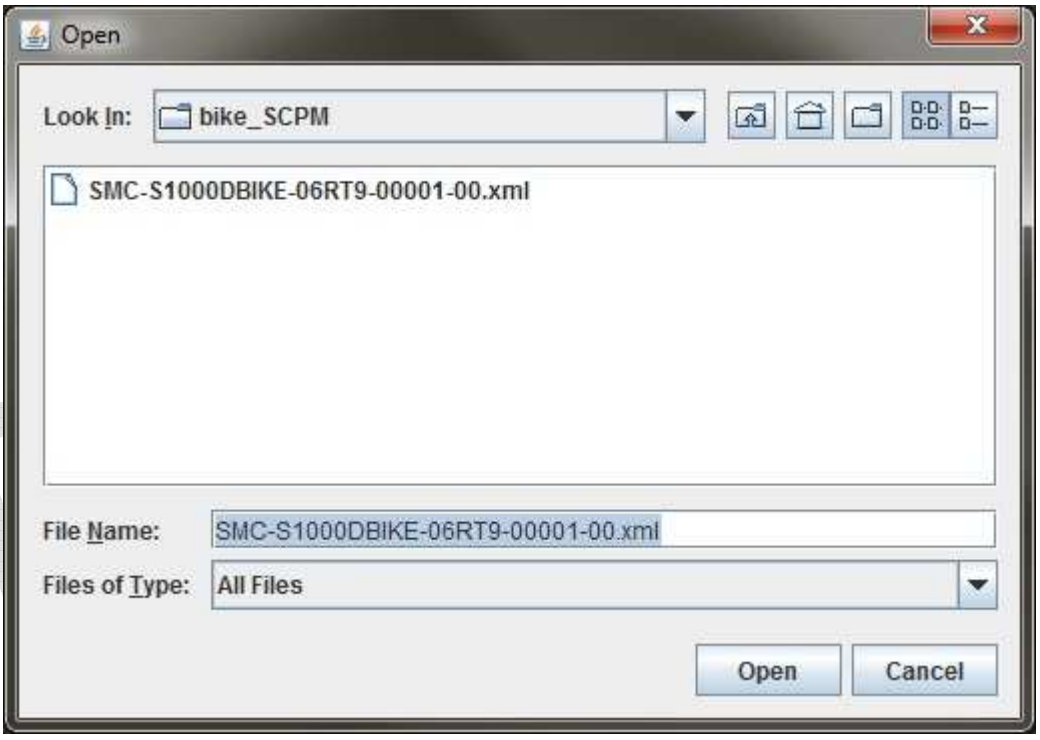

Figure 3.3b: Toolkit Search for SCPM

Step 3: Click the browse button to search for the Resource Package. Select the Resource Package and click Open. The following screenshot shows the sample Bike project Resource Package. See **Figure 3.3c**.

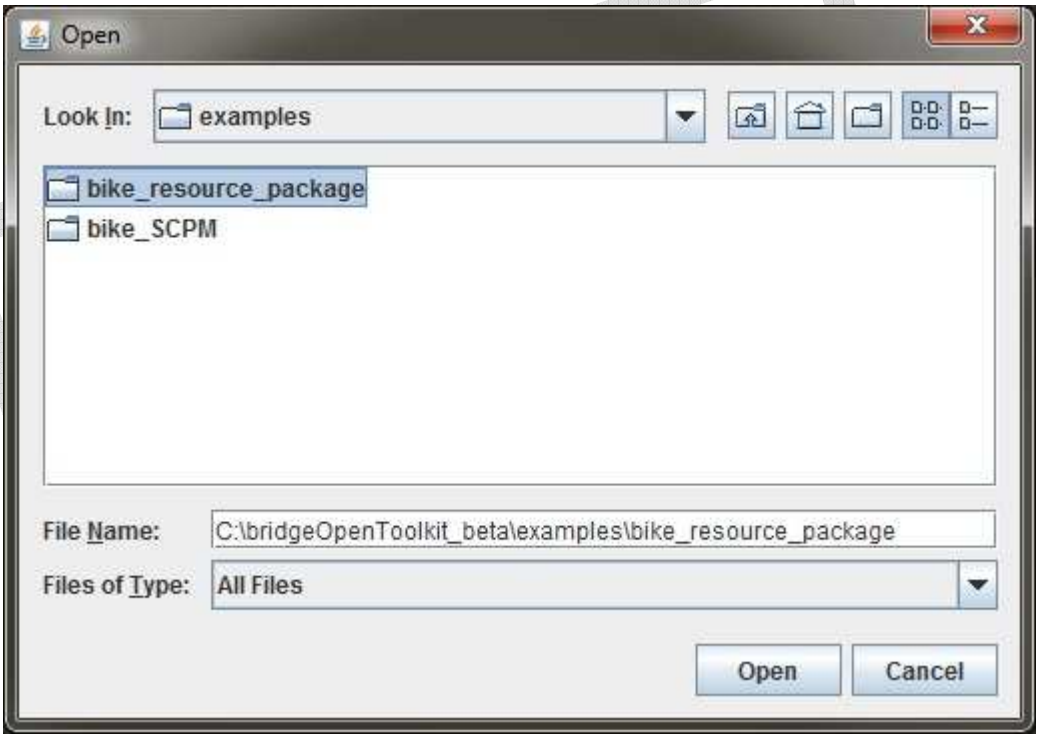

Figure 3.3c: Toolkit Search for Resource Package

Step 4: After the SCPM and Resource Package are selected, then click Run. The Toolkit will create the IMS Manifest and the content package. See **Figure 3.3d**.

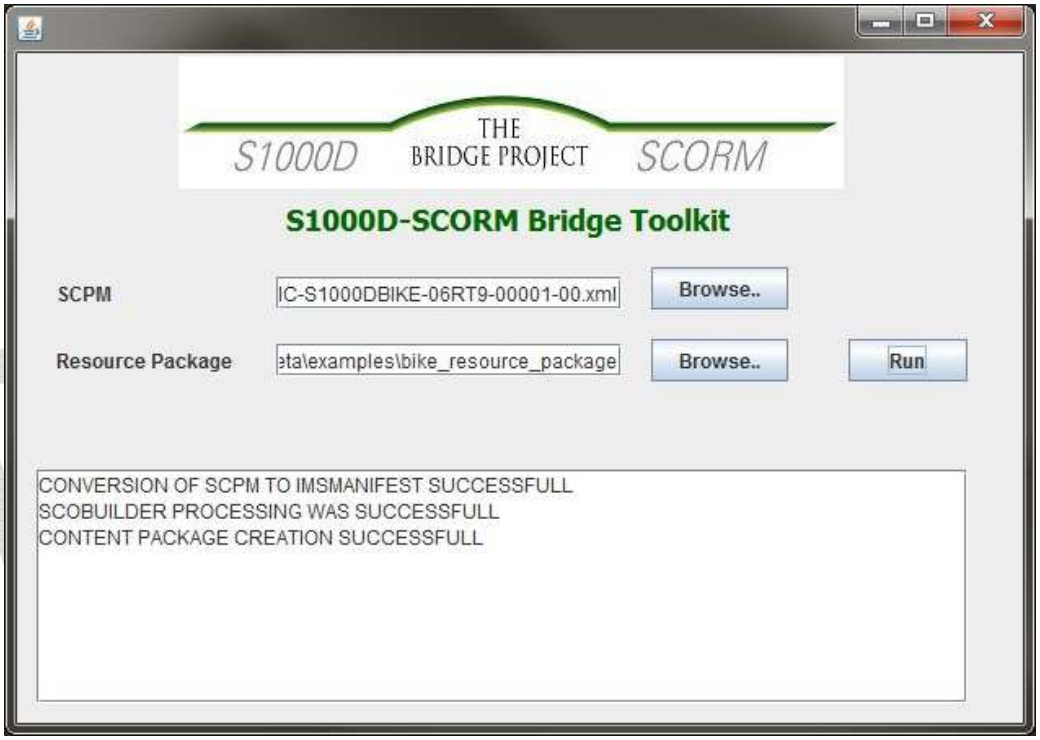

Figure 3.3d: Toolkit Successful Run

The Toolkit has now created a logs directory and SCORM 2004 Content Package zip file. The name of the zip file for the Bike project is S1000D_BIKE_Learning_Sample.zip. These files are found in the Toolkit directory.

3.4. How Developers Will Use the Toolkit

CSDB Vendors, LMS Vendors and Application Developers can use the Toolkit in three scenarios:

3.4.1. Publishing SCORM 2004 from an S1000D Compliant Application

In **Figure 3.4.1**, CSDB vendors produce S1000D applications that publish content that conforms to the S1000D standard. CSDB developers will be able to integrate this Toolkit into their code so that their product will be able to publish SCORM 2004 training content.

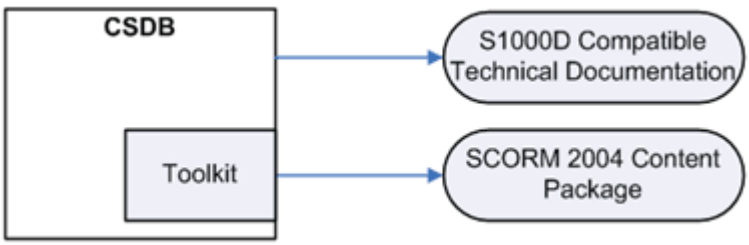

Figure 3.4.1: Publishing from CSDB

3.4.2. Importing S1000D Content into a Learning Management System

In **Figure 3.4.2**, LMS vendors produce applications that courseware developers use to assemble training content and publish in conformance with the SCORM 2004 standard. LMS vendors will be able to integrate this Toolkit into their code so that their product will be able to import S1000D content, and publish as SCORM 2004. In this usage, the LMS would be able to share, duplicate or modify the imported content to enhance existing courses, produce multiple courses and to conform to training design standards.

Figure 3.4.2: Importing S1000D Content into a Learning Management System

3.4.3. Independent Application Development

In **Figure 3.4.3**, an independent application can be created that will take the SCPM and associated resource package, from any S1000D application, as inputs, and output a SCORM 2004 Content Package that can be played on any Learning Management System that is certified to be compatible with the SCORM 2004 standard.

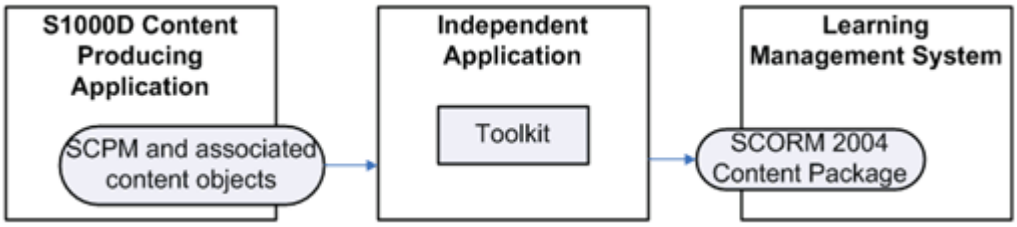

Figure 3.4.3: Independent Application Development

4. Using the Toolkit

4.1. The Input Package

The Toolkit requires two inputs:

- The SCPM: The SCPM is an XML file that is an output of S1000D, version 4.0. For the Bike demonstration project it is found in the directory, examples/bike_SCPM.
- The Resource Package directory: This directory contains the data module resource XML files referenced by the SCPM. For the Bike demonstration project it is the examples/bike_resource_package directory.

4.2. Toolkit Local files

The Toolkit contains two directories that are used to produce a SCORM Content Package: ViewApplication and xsd.

4.2.1. The ViewerApplication Directory

Figure 4.2.1 shows all the files required to render the data modules in the browser and to do the SCORM API communication. Examples include APIWrapper.js, s1000d_4.xslt, navScript.js and SCO header images.

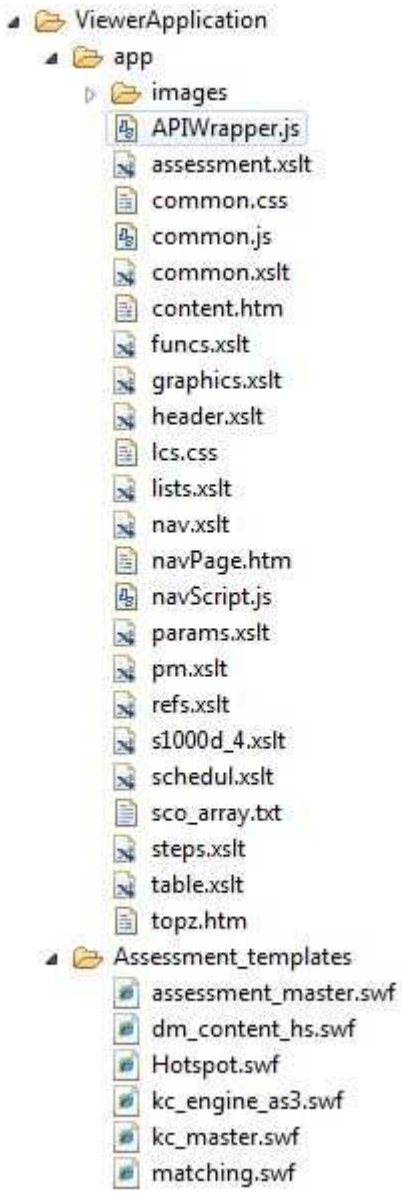

Figure 4.2.1: Viewer Application Files

4.2.2. The xsd Directory

Figure 4.2.2 shows all of the IEEE Learning Object Metadata (LOM) schemas files, the ADL SCORM 2004 3rd Edition schema files and IMS Content Package schema version IMS CP 1.1.4 (ADL and IMS schemas are highlighted).

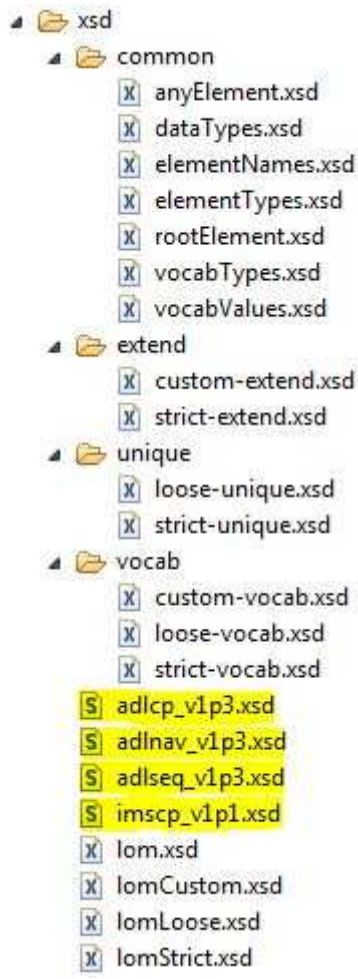

Figure 4.2.2: Schema Files

4.3. Running the Toolkit, Command Line

The Toolkit is also implemented into an application using command line interface. These are the instructions for running the Toolkit from the command line:

1. From the toolbar, click Start > Run.
2. In the Open field, type cmd.
3. Change the command prompt to c:\bridgeOpenToolkit0.9
4. Type java –jar Toolkit.jar “<scpm location>” “<resource package directory>” ex. java -jar Toolkit.jar "C:\bridgeToolkit0.9\examples\bike_SCPM\SMC-S1000DBIKE-06RT9-00001-00.xml" "C:\bridgeToolkit0.9\examples\bike_resource_package"

When the Toolkit processing completes successfully, the confirmation messages will be displayed:

- CONVERSION OF SCPM TO IMSMANIFEST SUCCESSFUL

- SCOBUILDER PROCESSING WAS SUCCESSFUL
- CONTENT PACKAGE CREATION SUCCESSFUL

4.4. Transformation

The Toolkit transformation process converts the content aggregation structure defined in the SCPM to the aggregation structure of a SCORM IMS Manifest file. All titles, metadata and referenced files from the SCPM are mapped to specific XML structures that define the IMS Manifest file.

The Toolkit locates the SCPM and resource folder by way of the user provided URL parameters. The SCPM and resource folder URLs allow the location of all the files needed to transform the S1000D managed content referenced in the SCPM to a SCORM Content Package. The Toolkit locates and gathers the content files into the folder structures referenced within the IMS Manifest. The content is also grouped within the IMS Manifest into SCO organizations to reflect the collection of SCOs defined in the SCPM.

The Toolkit transformation process also provides a Viewer Application (viewer) for the output SCORM Content Package. The viewer is a Web browser-based rendering environment. It consists of a collection XML Style sheet translation (XSLT) files along with other files necessary to provide content navigation and other features required for presentation of the content in a Web browser such as Internet Explorer. The viewer is required because the S1000D content is left in its original XML form. The Toolkit adds XML processing instructions to the S1000D XML content files to enable the viewer rendering operations.

The generated SCO files, all of the Viewer Application files and all of the necessary SCORM schema files are then added as references in the imsmanifest.xml structures to complete that transformation. The imsmanfiest.xml file and all of the required resources (S1000D data modules, S1000D ICN files, SCO files, Viewer Application files and the SCORM schemas) are then packaged into a .zip file to produce the output (SCORM 2004 3rd Edition Conformant Content Package).

Note: Currently the Toolkit does not perform validation on the on the Content Package. If there is a reference to a missing data module or ICN file in the SCPM or any of the referenced data modules then the imsmanifest.xml file produced could raise SCORM Content Package validation errors.

4.5. Reviewing the Output Package

The Toolkit consumes the inputs, processes the transformation procedures and produces a SCORM 2004 3rd Edition Conformant Content Package. It is placed inside the “bridgeToolkit0.9” directory. The name of the Content Package will be:

- <value of the scormContentPackageTitle element from the SCPM with _ replacing any whitespaces>.zip
- Example: S1000D_BIKE_Learning_Sample.zip

Note: If the provided Toolkit GUI is used to run the Toolkit then the confirmation messages or error messages will be saved in the directory "bridgeToolkit0.9/logs" to files named toolkit-<time stamp>.log. This feature does not currently available when the Toolkit.jar file is used to run the Toolkit.

When the Bike sample course is uploaded to a SCORM 2004 Conformant LMS and run, it will display the Bike course. **Figure 4.5** is a screenshot of one of the content screens.

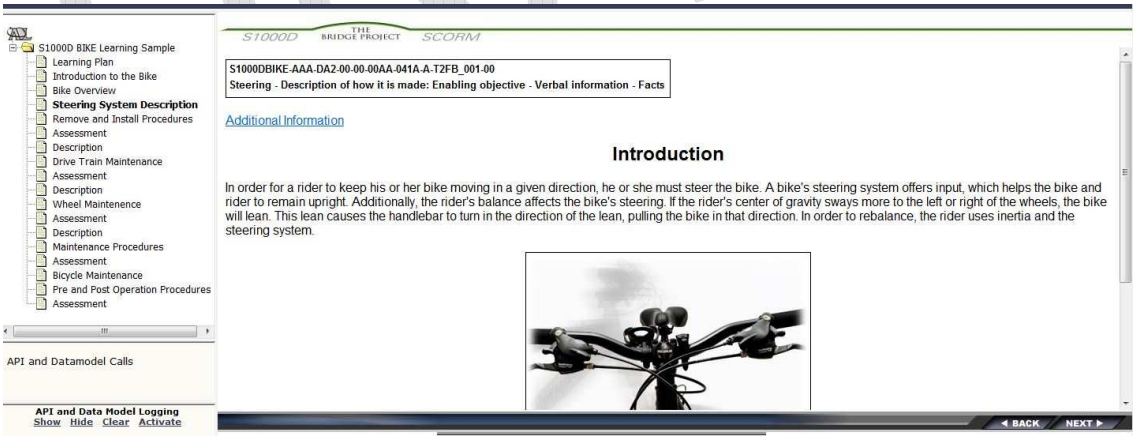

Figure 4.5: Bike Sample Course

5. Integrating the Toolkit

5.1. Conceptual Architecture Modules

The conceptual model is broken down into modules and sub modules that will allow for expandability and allow for plug-in support for future desired formats.

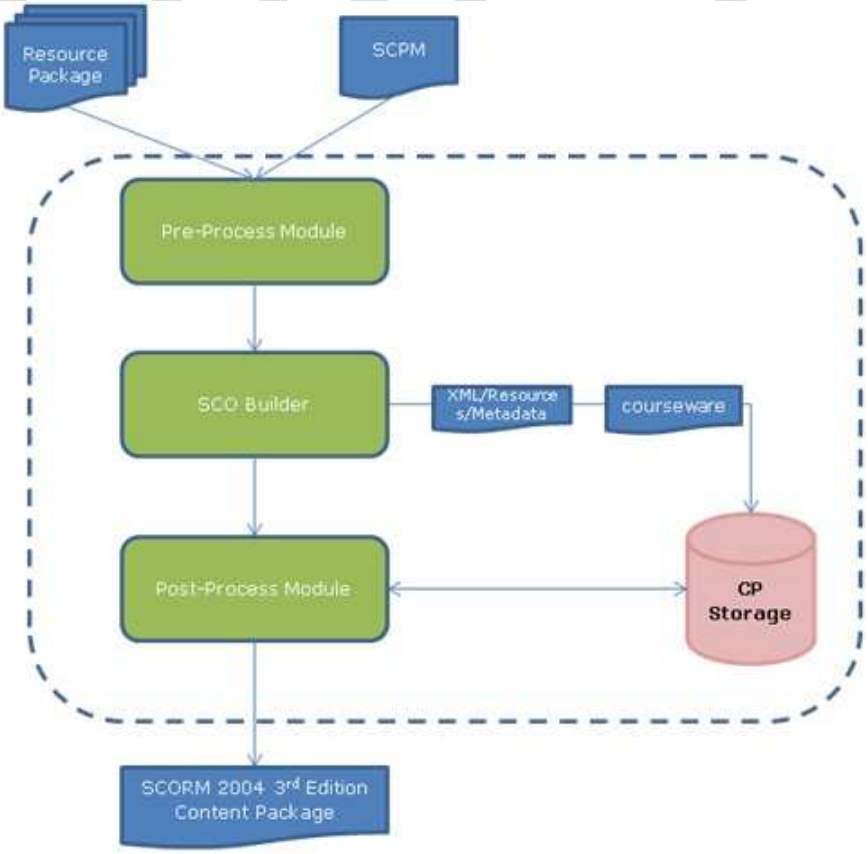

Figure 5.1: Toolkit Conceptual Architecture

5.2. Apache Commons Chain

The Apache Commons Chain supports the Toolkit's conceptual Architecture by providing an API that facilitates the "Chain of Responsibility" design pattern. For more information see: http://commons.apache.org/chain/

Each module in the Toolkit will be "command" that has an execute() method. Each execute() method is passed a "context" parameter that isolates the command

implementations from the environment. The Toolkit will utilize the "context" to manage the input required between modules. There are the default input key-value pairs that will be required for the Toolkit.

SCPM_FILE - String that represents the location of the S1000D SCPM file that is being converted into a SCORM Content Package.

RESOURCE_PACKAGE - String that represents the location of the resource package that contains all the S1000D files referenced in the SCPM to be included in the SCORM Content Package.

XML_SOURCE - The IMS Manifest file that is generated from the SCPM.

CP_PACKAGE - File that represents the directory that is being used to build the SCORM Content Package.

5.3. PRE PROCESS MODULE

Input: SCPM_FILE, RESOURCE_PACKAGE
Output: XML_SOURCE (The IMS Manifest file but will not have all the necessary information at this point to be a SCORM conformant.)
Description: The preprocessing is where the mapping from SCPM to SCORM IMS Manifest file begins (XSLT transform). The reason the for using a content aggregation structure that is a standard is important is so that the pre process can be easily swapped out to support other inputs in the future without affecting the rest of the Toolkit.

1. The first step takes all of the data module files and media files (ICN) in the RESOURCE_PACKAGE to create a URN Resource Map. In addition every media file (ICN) that is referenced in the data module is identified and located based on the <!ENTITY> elements in that data module. At this time the MEDIA_MAP will also be produced to be used in a later module to build the "resource/dependency" elements in the XML_SOURCE.

Figure 5.3a: Resource URN Map

The URN Resource Map is structured in this way:

```
<urn name="URN:S1000D:DMC-S1000DBIKE-AAA-D00-00-00-00AA-932A-T-H30A">
```

```
   <target type="figure">DMC-S1000DBIKE-AAA-D00-00-00-00AA-932A-T-
A_001-00_en-us.xml</target>
 </urn>
```

2. Next the SCPM_FILE is transformed with an XSLT file to create the XML_SOURCE.

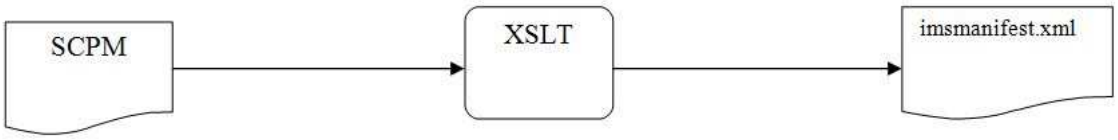

Figure 5.3b: Transformation

3. Then all of the files from the URN Resource Map are added to the XML_SOURCE as “resource” elements.

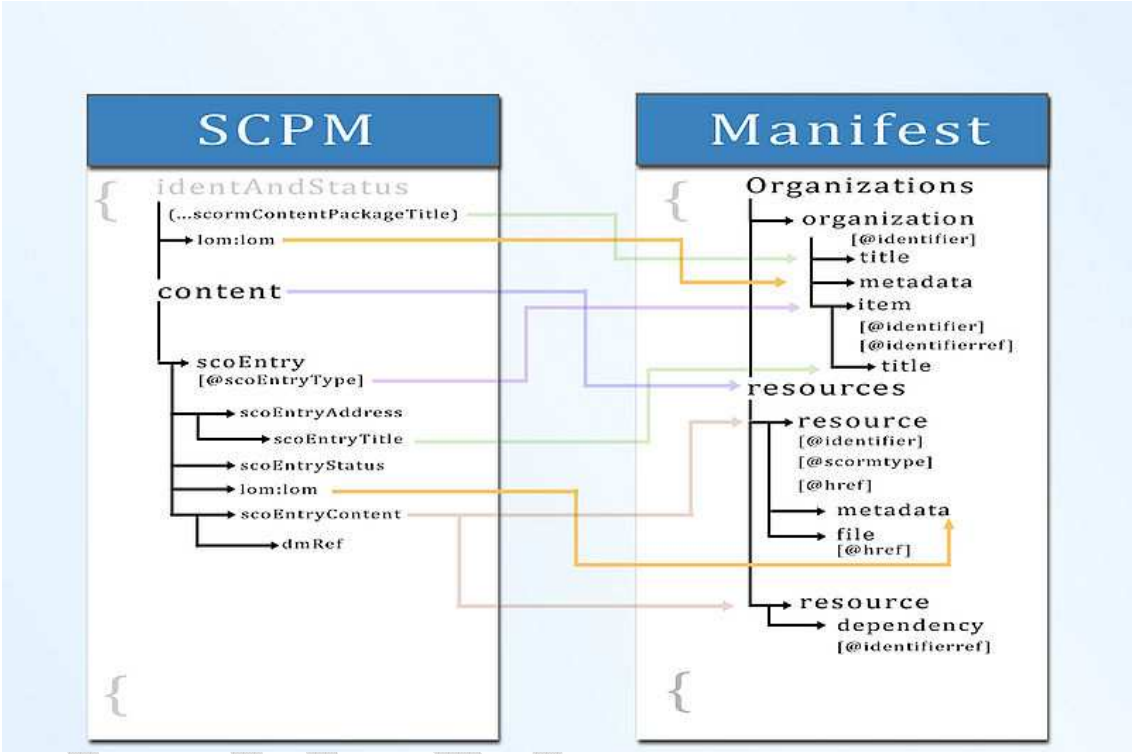

Figure 5.3c: SCPM - IMSMANIFEST Mapping

5.4. SCO BUILDER

Input: XML_SOURCE, CP_PACKAGE
Output: HTML SCOs, updated XML_SOURCE
Description: This is where the launchable learning resources are created. The default output of the Toolkit is HTML SCOs. The CP_PACKAGE is created during this module. The files from the resource package are copied over to the CP_PACKAGE and placed in a directory named "resources/s1000d". The Toolkit Viewer Application files (from Toolkit Local Files) are also copied over to the "resources/s1000d" directory. The created HTML SCOs will be copied over to the directory named "resources/scos".

The XML_SOURCE (IMS Manifest file) will be updated during this process. The location of the HTML SCOs will be used as the "href" attributes on the appropriate "resource" elements. The Viewer Application files will be added as a common asset "resource" element and a "dependency" element referencing this common resource will be added to all the "resource" elements that contain SCOs.

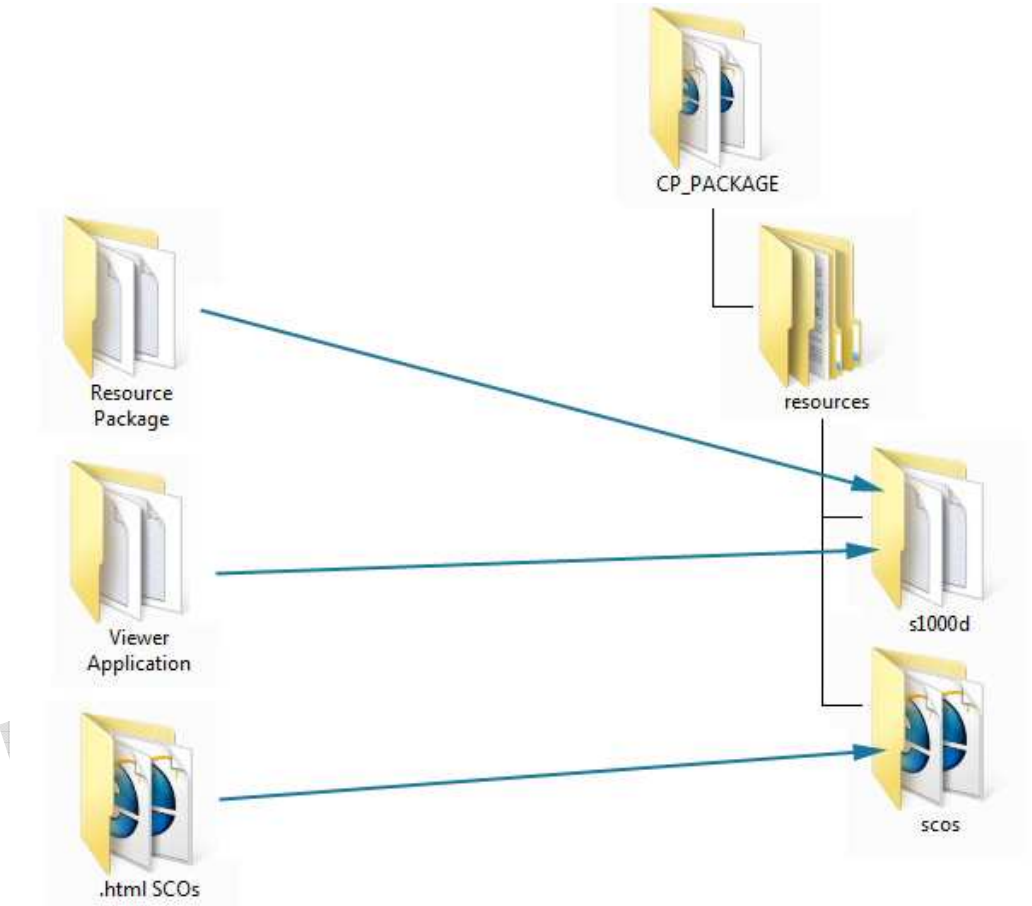

Figure 5.4: SCO Builder

5.5. POST PROCESS MODULE

Input: XML_SOURCE, CP_PACKAGE.
Output: SCORM 2004 3rd Edition Conformant Content Package
Description: The post process will copy the ADL Schemas, IEEE LOM Schemas and XML_SOURCE to the CP_PACKAGE and finally zip the file up into a SCORM Content Package. Any cleanup of the Toolkit can happen here as well.

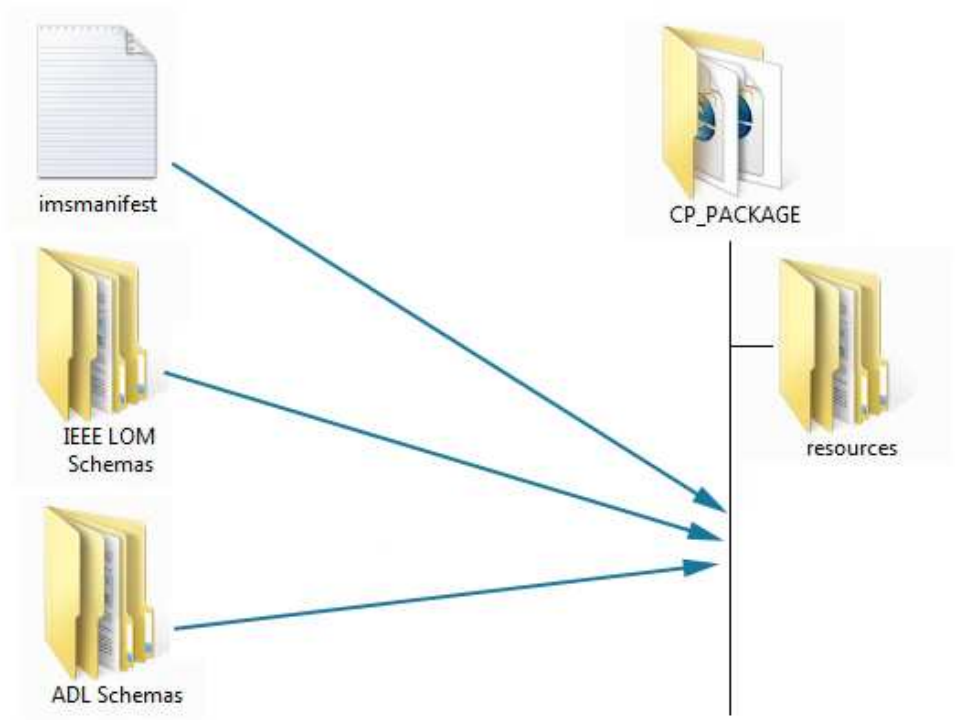

Figure 5.5: Post Process

5.6. VIEWER MODULE

Information on the View Module will be delivered with the final Toolkit product.

6. Troubleshooting

6.1. Log files

Log file data will be delivered with the final Toolkit product.

6.2. Error messages

Error messages and descriptions will be delivered with the final Toolkit product.

6.3. Debugging

Debugging instructions will be delivered with the final Toolkit product.

6.4. Sample files description

Descriptions of sample files will be delivered with the final Toolkit product.

7. Terms and Definitions

Term	Definition
Advanced Distributed Learning (ADL) Initiative	A DoD agency that leads the Federal participation with business and university groups, charged with developing consensus standards for training software and associated services. Source: http://www.adlnet.gov/About/Pages/adlinitiative.aspx.
Common Source Database (CSDB)	The CSDB is a collection of data modules. According to the S1000D specification, the CSDB is an information store and management tool for all objects required to produce the technical publications within projects. S1000D does not specify the design and implementation rules for a CSDB.
Data Module (DM)	The S1000D version 4.0 specification defines the Data Module in Chapter 3.2. Data Modules are small, reusable pieces of technical information. They are uniquely identified using a Data Module Code to allow ease of management and access in a database environment. Each data module contains a metadata section called the Identification and Status section which identifies and describes the data module and a Content section containing the technical information.
Extensible Stylesheet Language Transformation (XSLT)	Extensible Stylesheet Language Transformation (XSLT) is a declarative, XML-based language used for the transformation of XML documents. The original document is not changed; rather, a new document is created based on the content of an existing one. The new document may be serialized (output) by the processor in standard XML syntax or in another format, such as HTML or plain text. XSLT is most often used to convert data between different XML schemas or to convert XML data into web pages or PDF documents. Source: http://en.wikipedia.org/wiki/XSLT.
HTML SCOs	HTML SCOs are the starting points for the SCOs created by the Toolkit for the SCORM 2004 Content Package. These are created during the transformation process in the SCO Builder module.
Identification and Status	The identification and status section gives all the identification elements required to address and control an S1000D data module. It also provides the status elements for information on the security, quality and technical status together with the applicability of the overall data module content. Source: S1000D Technical Specification, Chapter 3.9.5.1.
IMS Manifest (imsmanifest.xml)	The IMS Manifest is a required XML file at the root of every SCORM package. It contains all organizational data, metadata and resource data for the SCORM course.

Learning Data Module (LDM)	S1000D supports technical training information development through the use of the learning data module. The Schema for this data module structures technical learning content and configures it to the system being instructed in the lessons. It also maintains the use of standard S1000D XML structures. By maintaining the common S1000D structures, reuse between technical data and its supporting learning content is possible without the need for conversion from other formats. Source: S1000D Technical Specification, Chapter 3.9.5.2.13.
Learning Management System (LMS)	A learning management system (commonly abbreviated as LMS) is a software application for the administration, documentation, tracking, and reporting of training programs, classroom and online events, e-learning programs, and training content. Source: http://en.wikipedia.org/wiki/Learning_management_system.
Learning Object Metadata (lom)	Learning Object Metadata is a data model, usually encoded in XML, used to describe a learning object and similar digital resources used to support learning. The purpose of learning object metadata is to support the reusability of learning objects, to aid discoverability, and to facilitate their interoperability, usually in the context of online learning management systems (LMS). Source: http://en.wikipedia.org/wiki/Learning_object_metadata.
Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD(AT&L))	The Office of the Undersecretary of Defense for Acquisition, Technology and Logistics (OUSD(AT&L)) is the title of a high-level civilian official in the United States Department of Defense. The Undersecretary of Defense for Acquisition, Technology and Logistics is the principal staff assistant and advisor to both the Secretary of Defense and the Deputy Secretary of Defense for all matters concerning the Acquisition, Technology and Logistics. This office has funded this Bridge project through its Reduction in Total Ownership Cost (RTOC) program. Source: http://en.wikipedia.org/wiki/Under_Secretary_of_Defense_for_Acquisition,_Technology_and_Logistics.
Organization	An organization is a level of training content hierarchy specified by the imsmanifest.xml.
Post Process Module	The Toolkit runs three modules to create the SCORM Content Package. The Post Process Module is the last of the modules to run. The Post Process Module copies the ADL Schemas, IEEE LOM Schemas and imsmanifest.xml (XML_SOURCE) to the SCORM Content Package (CP_PACKAGE) and finally zips the file up. Any cleanup of the Toolkit can happen here as well.
Pre Process Module	The Toolkit runs three modules to create the SCORM Content Package. The Pre Process Module is the first of the modules to run. The preprocessing is where the mapping from SCPM to SCORM manifest file begins (XSLT transform).

Project Management Committee	The Project Management Committee is composed of the Source Forge project team members.
Reduction in Total Ownership Cost (R-TOC)	The Reduction in Total Ownership Costs (R-TOC) is a Department of Defense- (DoD-) wide effort to reduce total ownership costs that grew out of numerous reviews and discussions at Program Executive Officers'/Systems Command (PEO/SYSCOM) Commanders' conferences, the Defense Science Board, and others. Source: http://ve.ida.org/rtoc/rtoc.html.
Resource	A resource is a physical element of a SCORM 2004 Content Package that is specified by the imsmanifest.xml.
Resource Package	A Resource Package is a directory of data modules, and associated ICNs, that are referenced by a SCORM Content Package Module.
S1000D	S1000D is an international specification for the procurement and production of technical publications. It covers the planning and management, production, exchange, distribution and use of technical documentation that support the life cycle of any civil or military project. Projects include air, land and sea vehicles or equipment. It is an SGML/XML standard for preparing, managing, and using equipment maintenance and operations information. See http://www.s1000d.org/.
S1000D SCORM Bridge Toolkit ("Toolkit")	The S1000D SCORM Bridge Toolkit is an open source tool that converts S1000D 4.0 data into SCORM 2004 content packages.
S1000D Technical Data Specification	The S1000D Technical Data Specification is a document which provides a detailed, technical description of the S1000D standard. See http://www.s1000d.org/.
SCO Builder Module	The Toolkit runs three modules to create the SCORM Content Package. The SCO Builder is the second of the modules to run. This is where the launchable learning resources are created. The output of the module is the set of html SCOs associated with each SCO. The CP_PACKAGE is created during this module.
scoEntry	The scoEntry is the XML element of the SCORM Content Package Module that identifies a module of training and maps to the SCO element of an imsmanifest.xml.
SCORM Content Package (CP)	The SCORM Content Package is the zip file that contains the training course created by the Toolkit. This output is SCORM 2004 conformant. This zip file is also known as a PIF file.
Sharable Content Object (SCO)	A Sharable Content Object (SCO) is the lowest level of interactive training content within a SCORM course. Since it is sharable, it can be re-used in any SCORM course. The IMS Manifest contains references to the SCOs of the course.

Sharable Content Package Module (SCPM)	The SCORM Content Package Module (SCPM) is a document in S1000D version 4.0 that identifies selected DMs for a SCORM course and maps DMs to the training objects of the course. Source: S1000D Technical Specification, Chapter 4.9.5.
Sharable Content Object Reference Model (SCORM)	The Sharable Content Object Reference Model (SCORM) is a collection of standards and specifications for web-based e-learning. It defines communications between client side content and a host system called the run-time environment, which is commonly contained within a learning management system. SCORM also defines how content may be packaged into a transferable content package. This Toolkit is based on SCORM 2004 3rd Edition. See http://www.adlnet.gov/Technologies/scorm.
Transformation	Transformation is the process of converting the aggregation structure of one file to another. The Toolkit uses transformation to convert the SCORM Content Package Module to the imsmanifest.xml.
Uniform Resource Name (URN) Map	A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) that uses the urn scheme, and does not imply availability of the identified resource. Both URNs (names) and URLs (locators) are URIs, and a particular URI may be a name and a locator at the same time. Source: http://en.wikipedia.org/wiki/Uniform_Resource_Name.

8. Points of Contact

The Points of Contact for the S1000D-SCORM Bridge Toolkit Project are:

- Tyler Shumaker, Toolkit Task Manager, shumaket@ctc.com
- Steve Worsham, Developer, Worsham.Steve@idsi.com
- Schawn Thropp, Bridge Project Manager, ThroppS@ctc.com
- Sean Rushing, Document Writer, srushing@inmedius.com
- Mike Hamerly, Document Writer, mhamerly@inmedius.com
- Wayne Gafford, Bridge Project Government Manager, wayne.gafford@adlnet.gov