

SCORM® 2004 3rd EDITION

Sharable Content Object Reference Model

Content Aggregation Model

NOVEMBER 16, 2006
VERSION 1.0



© 2006 Advanced Distributed Learning. All Rights Reserved.

このページは空白である。

Advanced Distributed Learning (ADL)

SCORM® 2004 3rd Edition

コンテンツアグリゲーションモデル (CAM) バージョン 1.0

ADLNet.gov から入手可能
(<http://www.adlnet.gov/>)

質問やコメントは ADL 問い合わせセンタ (ADLNet.gov) まで

このページは空白である.

日本語版訳者

eLC標準化推進委員会

石打智美
大仲 輝
仲林 清
本村孝則

日本電信電話 (株)
日立電子サービス (株)
放送大学
アベイズム (株)

このページは空白である.

チーフ テクニカル アーキテクト
Philip Dodds

テクニカル エディタ
Schawn E. Thropp

謝辞

ADLは、相互運用可能なeラーニングの標準および仕様の作成に関する下記の組織およびメンバーの継続的・献身的な協力に感謝したい。

**Alliance of Remote Instructional Authoring & Distribution
Networks for Europe (ARIADNE) (<http://www.ariadne-eu.org/>)**

Aviation Industry CBT Committee (AICC) (<http://www.aicc.org/>)

**Institute of Electrical and Electronics Engineers (IEEE)
Learning Technology Standards Committee (LTSC) (<http://ltsc.ieee.org/>)**

IMS Global Learning Consortium, Inc. (<http://www.imsglobal.org/>)

ADLはADLコミュニティに対してもSCORMの進化への貢献に対して感謝を表したい。

SCORM® 2004 3rd Edition documentation suite reprinted with permission from IEEE Std. 1484.11.1-2004 IEEE Standard for Learning Technology – Data Model for Content to Learning Management System Communication, Copyright 2004, by IEEE; IEEE Std. 1484.11.2-2003 IEEE Standard for Learning Technology – ECMAScript Application Programming Interface for Content to Runtime Services Communication, Copyright 2003, by IEEE; IEEE Std. 1484.12.1-2002 IEEE Standard for Learning Object Metadata, Copyright 2002, by IEEE; and IEEE Std. 1484.12.3-2005 IEEE Standard for Learning Technology – Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata, Copyright 2005, by IEEE. The IEEE disclaims any responsibility or liability resulting from the placement and use in the described manner.

SCORM® 2004 3rd Edition documentation suite reprinted with permission from IMS Content Packaging v1.1.4 Copyright 2004, by IMS Global Learning Consortium Inc. and IMS Simple Sequencing v1.0 Copyright 2003, by IMS Global Learning Consortium Inc. IMS Global Learning Consortium has made no inquiry into whether or not the implementation of third party material included in this document would infringe upon the intellectual property rights of any party. Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the document set forth in this document, and to provide supporting documentation to IMS. This material is being offered without any warranty whatsoever, and in particular, any warranty of non-infringement is expressly disclaimed. Any use of this material shall be made entirely at the implementer's own risk, and neither the IMS Global Learning Consortium, nor any of its members or submitters, shall have any liability whatsoever to any implementer or third party for any damages of any nature whatsoever, directly or indirectly, arising from the use of this material.

著作権

Copyright 2006 Advanced Distributed Learning Initiative (ADL). All rights reserved.

配給

本書の配給を許可するには以下の条件を満たす必要がある：

1. 本書および本書に言及されたイメージおよび事例の利用は、非営利目的、教育目的もしくは情報提供目的に限る。
2. 本書および本書に言及されたイメージおよび事例は、修正されずそのまゝの形であること。これには表紙および著作権、配給、再配布セクションも含まれる。

再利用

本書を全てまたは部分的に再利用するには以下の条件を満たす必要がある：

1. 再利用は、非営利目的で教育目的もしくは情報提供目的に限る。
2. 情報元として以下を適切に引用する

情報元：Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Content Aggregation Model Version 1.0, 2006.

著作権、配給および再利用に関するより詳細の情報および質問には、下記までお問い合わせ下さい：

ADL Co-Laboratory Hub
1901 North Beauregard Street, Suite 600
Alexandria, Virginia 22311
USA
703-575-2000

目次

セクション 1 SCORM コンテンツアグリゲーションモデル (CAM) 概要	1-1
1.1. SCORM コンテンツアグリゲーションモデル (CAM) ブック概説	1-3
1.1.1. SCORM コンテンツアグリゲーションモデル (CAM) ブックの内容は？	1-4
1.1.2. SCORM CAM ブックの利用	1-4
1.1.3. 他の SCORM ブックとの関係	1-5
1.2. SCORM コンテンツアグリゲーションモデル	1-7
セクション 2 SCORM コンテンツモデル	2-1
2.1. SCORM コンテンツモデル要素	2-3
2.1.1. アセット	2-3
2.1.2. 共有可能コンテンツオブジェクト (SCO)	2-3
2.1.3. アクティビティ	2-5
2.1.4. コンテンツオーガニゼーション	2-6
2.1.5. コンテンツアグリゲーション	2-7
セクション 3 SCORM コンテンツパッケージング	3-1
3.1. コンテンツパッケージング概要	3-3
3.2. コンテンツパッケージ要素	3-4
3.2.1. パッケージ	3-5
3.2.2. マニフェスト	3-5
3.2.3. パッケージ交換ファイル (PIF)	3-6
3.3. マニフェストの要素	3-7
3.3.1. メタデータ	3-7
3.3.2. オーガニゼーション	3-7
3.3.3. リソース	3-14
3.3.4. コンテンツ	3-15
3.4. コンテンツパッケージの作成	3-16
3.4.1. マニフェストファイル	3-16
3.4.2. コンテンツパッケージマニフェストの拡張	3-41
3.4.3. コンテンツパッケージマニフェストの Href の扱い	3-42
3.5. SCORM コンテンツパッケージアプリケーションプロファイル	3-47
3.5.1. リソースパッケージ	3-47
3.5.2. コンテンツアグリゲーションパッケージ	3-50
3.5.3. SCORM コンテンツパッケージアプリケーションプロファイル要件	3-52
3.6. ベストプラクティスおよび実践的なガイドライン	3-54
3.6.1. 単一コースに対する複数オーガニゼーション	3-54
3.6.2. 再利用のための学習コンテンツのパッケージング	3-54
3.6.3. <dependency> 要素の利用	3-54
セクション 4 SCORM メタデータ	4-1
4.1. SCORM メタデータ概要	4-3
4.2. LOM メタデータ作成	4-5
4.2.1. <lom> 要素	4-7
4.2.2. <general> 要素	4-9
4.2.3. <lifeCycle>要素	4-18
4.2.4. <metaMetadata> 要素	4-24

4.2.5.	<technical>要素	4-33
4.2.6.	<educational> 要素	4-45
4.2.7.	<rights>要素	4-55
4.2.8.	<relation>要素	4-58
4.2.9.	<annotation>要素	4-64
4.2.10.	<classification>要素	4-67
4.2.11.	共通データタイプ	4-75
4.3.	LOM XML スキーマバリデーションアプローチ	4-80
4.3.1.	厳格なスキーマバリデーションアプローチ	4-80
4.3.2.	カスタムスキーマバリデーションアプローチ	4-81
4.3.3.	柔軟なスキーマバリデーションアプローチ	4-81
4.4.	メタデータ拡張	4-82
4.4.1.	データ要素拡張	4-83
4.4.2.	語彙拡張	4-83
4.5.	メタデータと SCORM コンテンツモデル要素	4-85
4.5.1.	コンテンツアグリゲーション記述メタデータ	4-86
4.5.2.	コンテンツオーガニゼーション記述メタデータ	4-87
4.5.3.	アクティビティ記述メタデータ	4-88
4.5.4.	SCO 記述メタデータ	4-89
4.5.5.	アセット記述メタデータ	4-89
セクション 5	SCORM シーケンシングおよび プレゼンテーション	5-1
5.1.	シーケンシングおよびプレゼンテーション	5-3
5.1.1.	<sequencing>要素	5-3
5.1.2.	<controlMode>要素	5-5
5.1.3.	<sequencingRules> 要素	5-7
5.1.4.	<limitConditions>要素	5-15
5.1.5.	<auxiliaryResources> 要素	5-16
5.1.6.	<rollupRules>要素	5-17
5.1.7.	<objectives>要素	5-24
5.1.8.	<randomizationControls>要素	5-31
5.1.9.	<deliveryControls> 要素	5-33
5.1.10.	<constrainedChoiceConsiderations> 要素	5-34
5.1.11.	<rollupConsiderations> 要素	5-36
5.1.12.	<sequencingCollection>要素	5-38
5.2.	プレゼンテーション/ナビゲーション情報	5-40
5.3.	コンテンツパッケージングとの関係	5-43
付録 A	略語表	A-1
略語表		A-3
付録 B	参考文献	B-1
参考文献		B-3
付録 C	ドキュメント改定履歴	C-1
ドキュメント改定履歴		C-3

セクション 1

SCORM コンテンツアグリゲーションモデル (CAM) 概要

このページは空白である。

1.1. SCORM コンテンツアグリゲーションモデル (CAM) ブック概説

SCORM は、しばしば本棚の一組の本に例えられる。コンテンツアグリゲーションモデル (CAM) ブックは、その一組の本の一冊である。(図 1.1.a: SCORM ブックシェルフの 1 部としてのコンテンツアグリゲーションブック参照)。他のブックおよびブック間の関係についてのより詳細な情報は SCORM2004 概要に記述されている。SCORM CAM ブックは、学習活動で用いられる要素、これらの要素を一つのシステムから他のシステムに交換するためにどのようにパッケージするか、検索と発見を可能とするためにこれらの要素をどのように記述するか、これらの要素に対してどのようにシーケンシング情報を記述するかを説明している。CAM は学習コンテンツの一貫性のある保存、ラベル付け、パッケージング、交換、発見を支援する。

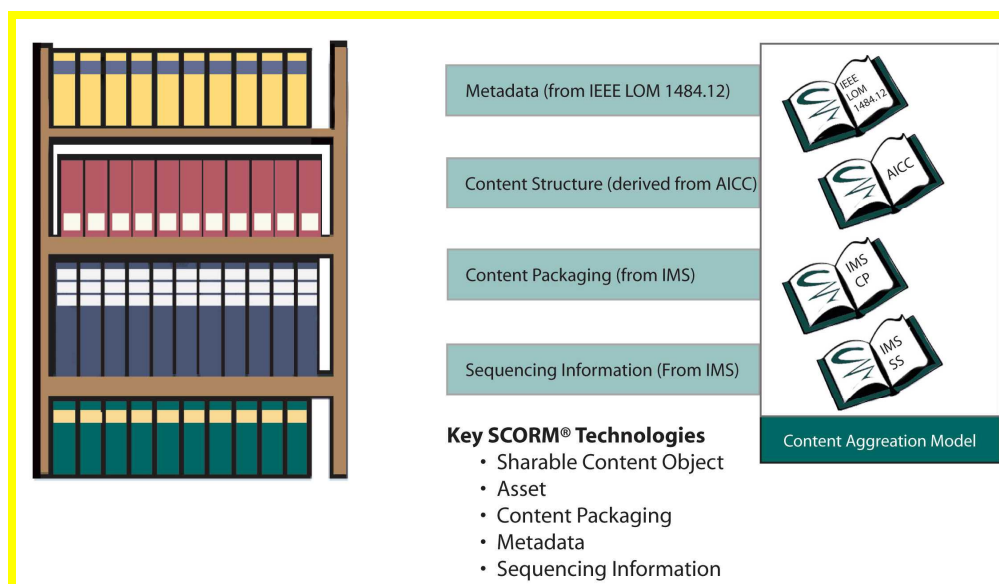


図1.1a: SCORM ブックシェルフの 1 部としての SCORM コンテンツアグリゲーションモデルブック

1.1.1. SCORM コンテンツアグリゲーションモデル (CAM) ブックの内容は？

SCORM コンテンツアグリゲーションモデル (CAM) ブックでいくつかの主要なコンセプトが導入される。このブックは、コンテンツおよびコンテンツオーガニゼーション（コース、レッスン、モジュールなど）を構築する要件および役割を説明する。このブックは、コンテンツパッケージの作成、コンテンツパッケージの要素へのメタデータの適用、コンテンツパッケージのコンテキストでのシーケンシングおよびナビゲーション詳細の適用に関する情報を含む。SCORM コンテンツパッケージングは、このブックに記述されるように、コンテンツ構成、学習コンテンツ、コンテンツ構成の各種の要素を記述するメタデータ、そして、シーケンシングやナビゲーションルールを説明するのに一貫性のある手段を提供する。この一貫性はコンテンツパッケージおよびそれらのリソースの検索および発見を可能にし、異なるシステムで同じように動作するコンテンツオーガニゼーションを構築し、コンテンツパッケージのコンテンツの標準的な理解をすることを可能にしている。包括的に纏められた議題は以下の通り：

- コンテンツモデル： CAM ブックを通して利用される共通用語の定義
- コンテンツパッケージング： 学習コンテンツを作成し統合する要件と説明
- メタデータ： SCORM 要素の記述に対する要件および説明
- シーケンシングおよびナビゲーション： シーケンシングおよびナビゲーション情報を定義する要件と説明

1.1.2. SCORM CAM ブックの利用

このブックは編集、オーサリングツールベンダ、コンテンツ開発者もしくは以下を作成、編集しようとする者すべてに役に立つ：

- SCORM コンテンツモデル要素（アセット、SCO、アクティビティおよびコンテンツオーガニゼーション）、
- SCORM コンテンツパッケージ（シーケンシングおよびナビゲーション情報のあるなしに関わらず）もしくは
- SCORM メタデータ

このブックは、コンテンツパッケージを読み込むシステムを開発する者を支援する。コンテンツパッケージをどのように処理するかを記述するこのブックを通じて、様々な要件が定義されている。

本ブックの初めの部分の「セクション 1：SCORM コンテンツアグリゲーションモデル (CAM) 概要」、「セクション 2：SCORM コンテンツモデル」では、一般的な CAM 関連の概念を取り扱っている。これらのセクションは SCORM CAM の背景にある概念の概要を知りたい人および技術的な詳細を深堀りしたくない人に勧められる。他にこれらのセクションが役に立つと思われる人々は、SCORM CAM の変更点について知りたい人などが含まれる。「セクション 2.1.3：コンテンツオーガニゼーション」は、例えば、アクティビティがどのように SCORM CAM に影響を与えるかについて記述している。

「セクション 3：SCORM コンテンツパッケージング」は、CAM 固有の技術的な詳細が提供される最初のセクションである。このセクションはマニフェスト、コンテンツパッケージ、SCORM コンテンツオーガニゼーションパッケージアプリケーションプロファイル、SCORM リソースパッケージアプリケーションプロファイル、および、ベストプラクティスおよび実践的なガイドラインを説明している。このセクションは SCORM コンテンツパッケージの様々な個別要素に関する技術的

な詳細を取り扱うだけでなく、どのようにコンテンツパッケージを組み立てるかに関しても、マニフェストの1部を抜粋して説明する。

「セクション4:」では、分類のためにメタデータを作成する全ての観点を取り上げる。これには Learning Object Metadata (LOM) Extensible Markup Language (XML) 検証アプローチ、メタデータ拡張、および、メタデータアプリケーションプロファイルが含まれる。このセクションはコンテンツパッケージの SCORM コンテンツモデル要素にメタデータをどのように関連付けるかを説明している。

「セクション5: SCORM シーケンシングおよびナビゲーション」では、シーケンシングおよびナビゲーション情報の導入が SCORM コンテンツアグリゲーションモデルに影響する方法を取り上げている。このセクションは、シーケンシングおよびナビゲーション情報を XML でどのように作成するか、および、これらのルールをコンテンツパッケージマニフェストにどのように設定するかを説明している。このセクションは、シーケンシング戦略要求を表す XML を作成する要件を記述する。

1.1.3. 他の SCORM ブックとの関係

それぞれの SCORM ブックは単体でも活用できるように意図されている一方で、内容が互いに重複する部分がある。例えば、本ブックは、SCO やアセットといった SCORM コンテンツの要素に主に焦点を当てるが、これらのオブジェクトは SCORM 対応 LMS に起動されるため、コンテンツ起動については、SCORM ランタイム環境ブック[2]で何度も言及されている。

同様に、SN ブックは、SCORM シーケンシングおよびナビゲーションプロセスの詳細を取り上げ、これには、LMS がナビゲーション要求および関連するアクティビティをどのように評価するかの詳細が含まれるが、本ブックは SN ブックに説明されているシーケンシング情報を含むマニフェストを取り上げるため、シーケンシングおよびナビゲーションの基礎のいくつかに簡単に触れている。

重複する部分を整理するのに、「セクション1.1.3.1: SCORM ランタイム環境ブック」および「セクション1.1.3.2:」でこれらのブックの内容の簡単な説明を提供している。

1.1.3.1. SCORM ランタイム環境ブック

SCORM ランタイム環境の目的は、SCO と LMS 間の相互運用性に関する手段を提供することである。SCORM は、コンテンツを作成したツールに関係なく学習コンテンツが複数の LMS で相互運用可能となる手段を提供する。そのためには、コンテンツを起動する共通の方法、コンテンツが LMS と交信する共通の方法、そしてコンテンツ実行時に LMS とコンテンツ間で交換されるあらかじめ決められたデータ要素が必要である。SCORM ランタイム環境の3つのコンポーネントは、本書では、起動、API(Application Program Interface)、データモデルと定義されている。これらの要素の技術的な詳細は SCORM RTE ブックで説明されているが、RTE のこれらの要素のそれぞれの簡単な概要は以下の通りである。

起動は、すべての SCORM 対応コンテンツは、学習者へ配信および表示するのに SCORM 対応 LMS に依存するように、LMS と SCORM コンテンツ間の関係を定義する。さらには LMS はどの SCORM コンテンツが次に配信されるべきかを決定する役割がある。この新たな役割は SCORM SN ブックで説明されているが、SCORM RTE ブックでも触れられている。

SCORM API は、SCORM RTE ブックで説明される通り、LMS と LMS が立ち上げる SCO 間の交信を可能にするために、LMS ベンダとコンテンツオーサリングツールベンダ両者に合意された予め定義

された機能を提供する。これらの機能は、LMS と LMS が起動する SCO の間のハンドシェイクを確立する手段を提供して、起動プロセスを完結させ、また、SCO が終了するとそのハンドシェイクを解放する。さらに、これらの機能は SCORM コンテンツがアセスメント結果などのデータを LMS に設定、および取得することを可能にし、また、このプロセスの最中に発生したエラーをチェックすることも可能にする手段を提供する。

SCORM ランタイム環境データモデルは、SCORM RTE ブックで説明されている通り、データを LMS から取得および LMS に設定するために利用されるデータ要素を提供する。例えば、学習者からのテスト点数を伝達するとき、LMS に学習者がどのような結果を出したか知らせるために、SCO は `cmi.score.scaled` と呼ばれる SCORM データモデル要素を利用する。これとその他全ての SCORM データモデル要素は、SCORM RTE ブックに詳しく説明されている。

SCORM CAM で説明されている様々なコンセプトが SCORM RTE に影響を与える。コンテンツ・パッケージ・マニフェスト内で定義されたデータは、いくつかの SCORM ランタイム環境データモデル要素の初期値に影響を与える。マニフェストからのデータはコンテンツを学習者へ配信および起動する過程で利用され、RTE に影響を与える。これらの関係および他の関係が SCORM CAM ブックを通して説明されている。

1.1.3.2. SCORM シーケンシングおよびナビゲーションブック

SCORM SN ブックは、IMS シンプルシーケンシング (SS: Simple Sequencing) 仕様バージョン 1.0 に基づく。IMS SS 仕様は、どんな LMS でも一貫性のある方法で分割された学習アクティビティをシーケンスするという作成された学習経験の意図された動作を表す方法を定義する。

SCORM SN モデルは、どのように IMS SS が SCORM 環境で適用され、拡張されるかを定義する。SCORM 対応 LMS がランタイム時にシーケンシング情報を処理するために、実装しなければならない必要な動作と機能性を定義する。より具体的には、起動されたコンテンツオブジェクトおよび作成されたシーケンシング戦略と学習者との相互作用の結果に基づき、アクティビティツリーにおける学習アクティビティの分岐とフローを記述する。アクティビティツリーは、LMS が各学習者に対して管理する学習アクティビティの概念的構造である。

SCORM SN ブックは、学習者やシステム主導によるナビゲーションイベントがどのように発生し、処理されるのか、そして、結果として配信される学習アクティビティがどのように決められるかを説明している。配信されることが決められた各学習アクティビティは、関連したコンテンツオブジェクトを持っている。SCORM RTE モデルは、配信されるコンテンツオブジェクトがどのように起動されるかを記述している。ある学習者とコンテンツ構成にとって、起動したコンテンツオブジェクトのシーケンスは、学習活動（コンテンツオブジェクトとの学習者の相互作用）を提供する。SCORM RTE モデルは、LMS が学習活動の結果をどのように管理するか、および、その学習活動がアクティビティツリーにどのように影響するかを記述する。

SCORM CAM ブックで説明された様々なコンセプトは SCORM SN ブックに関係がある。CAM は、どのようにシーケンシングルールを作るかおよびどのようにそれらのルールを XML で表現するかを説明する。次に、CAM は、これらのシーケンシング情報を適用するのにどのように既存マニフェストに載せるかを説明する。XML バインディングのシーケンシング情報とこの情報のプロセスおよび動作の間の関係の詳細については SCORM SN ブックを参照されたい。

1.2. SCORM コンテンツアグリゲーションモデル

SCORM コンテンツアグリゲーションモデルは、学習の設計者および開発者が、望ましい学習経験を配信目的として学習リソースを統合するための、学習分類に対して中立的な手段を表現する。学習リソースは、学習活動で利用される情報の表現である。学習活動は電子的もしくは非電子的リソースによってサポートされるアクティビティから構成される。

学習経験の作成および配信プロセスの一つのアクティビティは、単純なアセットをより複雑な学習リソースに作成、配信および統合し、これらのリソースを予め定義された配信のシーケンスに組み立てることである。SCORM コンテンツアグリゲーションモデルはこのプロセスをサポートし以下より成り立つ：

- コンテンツモデル： 学習経験のコンテンツ構成要素を定義する名称
- コンテンツパッケージ： 学習経験の意図された動作をどのように表すか（コンテンツモデル）、および、異なった環境間の移動に対して学習リソースのアクティビティをどのように集約するか（コンテンツパッケージング）を定義する
- メタデータ： コンテンツモデルの部品の特定のインスタンスを記述するメカニズム

シーケンシングおよびナビゲーション： 意図されたシーケンスおよびアクティビティの順序を記述するルールを定義するモデル。アクティビティは学習者に配信されるのに、学習リソースを参照することもあれば、ないこともある

このページは空白である。

セクション 2

SCORM コンテンツモデル

このページは空白である。

2.1. SCORM コンテンツモデル要素

SCORM コンテンツモデルは学習リソースから学習経験を作成するのに利用される SCORM 要素を記述する。コンテンツモデルはこれらの下位レベルの共有可能な学習リソースが、どのように上位レベルの教材単位に集約されるかを定義する。SCORM コンテンツモデルはアセット、共有可能コンテンツオブジェクト (SCO) およびコンテンツオーガニゼーションから成り立つ。

2.1.1. アセット

学習リソースの最も基本的な形はアセットである。アセットは、メディアを電子的に表現したもので、Web クライアントによって実現され学習者に提示される、テキスト、イメージ、音、アセスメントオブジェクトもしくはその他のデータの電子的表現である (図 2.1.1a 参照)。一つ以上のアセットが集まって他のアセットを作るように収集される。いくつかのケースでは、アセットは学習経験の 1 部として起動される。

アセットはレポジトリ内で検索と発見を可能にするためにアセットメタデータ (下記のアセットメタデータ定義を参照) で記述され、再利用可能となる。アセットをアセットメタデータと関連させる手段はコンテンツパッケージである (セクション 3: SCORM コンテンツパッケージ)。

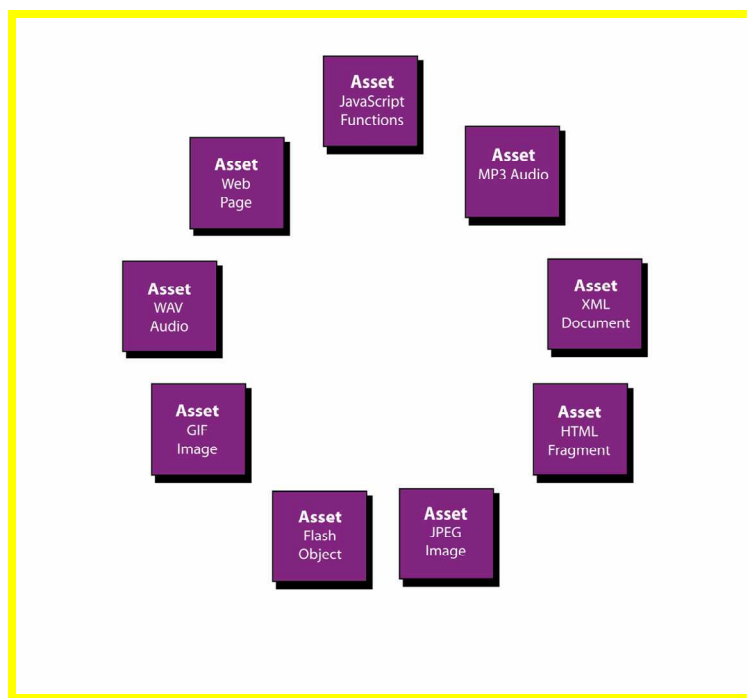


図 2.1.1a: アセットの例

2.1.2. 共有可能コンテンツオブジェクト (SCO)

SCO は、1 つもしくは複数のアセットが集まったもので LMS と通信するのに SCORM RTE を利用する単一の起動可能な学習リソースである。SCO は SCORM ランタイム環境データモデルを利用して LMS によって追跡される学習リソースの最小レベルの塊を表す。SCO とアセットの唯一の違いは、SCO が IEEE ECMA Script Application Programming Interface for Content to Runtime Services Communication ドラフト規格[1]を利用して LMS と通信することである。以下の図 2.1.2a はいくつのアセットから成る SCO の例を図示する。

再利用可能性を向上させるためには、SCO は自身の学習コンテキストに依存しないほうがよい。例えば、SCO は異なった学習目標を達成するために異なった学習経験で再利用することができる。さらに、アクティビティ（コンテンツオーガニゼーション参照）は、上位の学習目標を達成するための上位レベルユニットの教材もしくはトレーニングを作るのに一つ以上の SCO リソース（および/もしくはアセット）を統合する。

SCO は、複数の学習文脈で再利用が可能であるような小さな単位として意図されている。SCORM は SCO の厳密なサイズを制限してはいない。コンテンツ設計およびアクティビティ編集時、SCO のサイズを決定するときに、ランタイム時に LMS によって追跡されるコンテンツの論理的な最小サイズが考慮されるべきである。オーガニゼーションへ再利用するためには SCO のサイズを考慮すべきである。SCO のサイズの決定に影響を与える可能性のある他の要因は、学習結果を達成するのにどれだけの情報が要求されるか、またシーケンシングの分岐を決定する箇所などが含まれる。

SCO はレポジトリ内で検索と発見を可能にするために SCO メタデータ（下記の SCO メタデータ定義参照）で記述され、再利用可能となる。SCO を SCO メタデータと関連させる手段はコンテンツパッケージである（セクション 3: SCORM コンテンツパッケージ）。

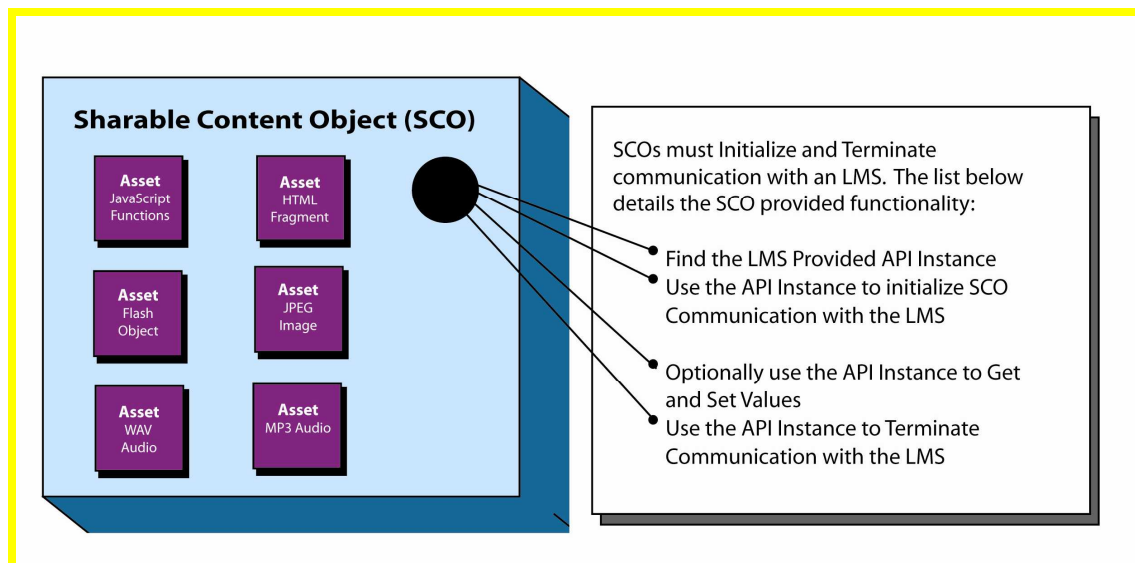


図2.1.2a: 共有可能なコンテンツオブジェクト (SCO)

SCO は SCORM ランタイム環境で定義された要件を遵守するように要求されている [2]。これは、SCO が、LMS が提供した API インスタンスを発見する手段を持たなければならないこと、および最小限の API メソッド (Initialize(“ ”) and Terminate(“ ”)) を呼び出さなければならないことを意味する。他の API メソッドは、オプションでありコンテンツの性質に依るので、呼び出さなくてもよい。

SCO が SCORM RTE を利用しなければならないが、そのために以下のメリットをがもたらされる：

- SCORM RTE をサポートするいかなる LMS も、誰が作成したかにかかわらず、SCO を起動し追跡できる
- SCORM RTE をサポートするいかなる LMS も、SCO を追跡し、いつ SCO が開始され、いつ終了したかを知ることができる
- SCORM RTE をサポートするいかなる LMS もどんな SCO を同じ方法で起動することができる

2.1.3. アクティビティ

学習アクティビティは、おおまかには、意味のある教示の単位ということができる。それは、概念的には、課程を進捗する間に学習者が行うことを指している。学習アクティビティは、学習リソース（SCO またはアセット）を学習者に提供するか、あるいは、複数のサブ学習アクティビティから構成される。

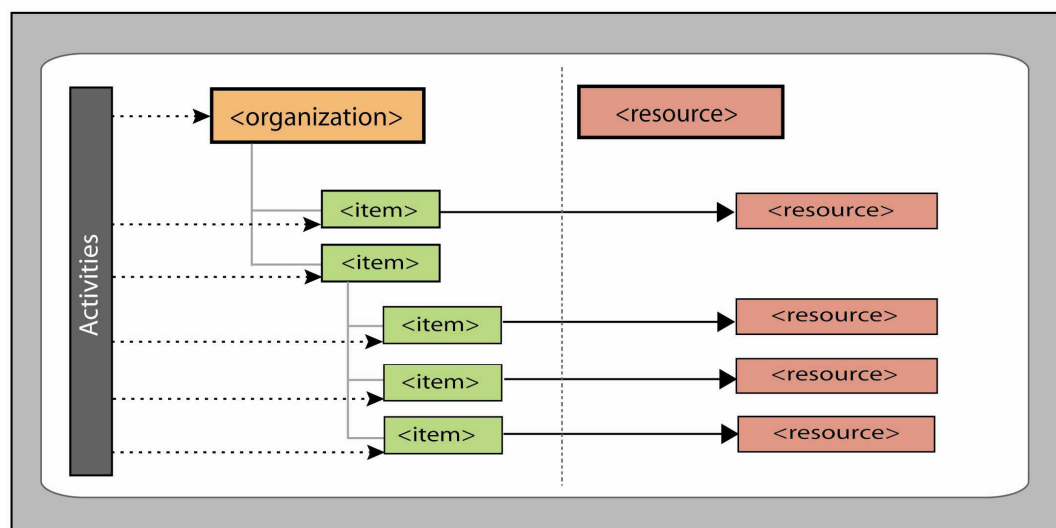


図2.1.3a アクティビティの概念的表現

コンテンツオーガニゼーションの中で表現されるアクティビティは他のアクティビティ（サブアクティビティ）から構成される場合があり、サブアクティビティはさらに他のアクティビティから構成されることがある。アクティビティのネストのレベルには決められた上限はない。特定の学習用語（例えば、コース、章、モジュール、など）がアクティビティの階層レベルに対応付けられている場合があるが、これは必要条件ではない。他のアクティビティから構成されないアクティビティ（葉アクティビティ）は、そのアクティビティを実行するための学習リソース（SCO またはアセット）と対応付けられる。

他のアクティビティから構成されるアクティビティはクラスタとも呼ばれる。アクティビティおよびクラスタに対して、どのように動作が定義されるかについては、SCORM SN ブックを参照されたい。

コンテンツオーガニゼーション中の各アクティビティには、リポジトリにおける検索・発見のためのメタデータを付与でき、これによって再利用の機会を設け、メンテナンスを容易にすることができる。

2.1.4. コンテンツオーガニゼーション

コンテンツオーガニゼーションは、構造化された教示の単位（アクティビティ）において、コンテンツの利用方法の意図を定義した表現ないしマップである。このマップは、アクティビティが他のアクティビティとどのように関連しているかを示している。以下の図 2.1.4a はコンテンツオーガニゼーションの例を示している。

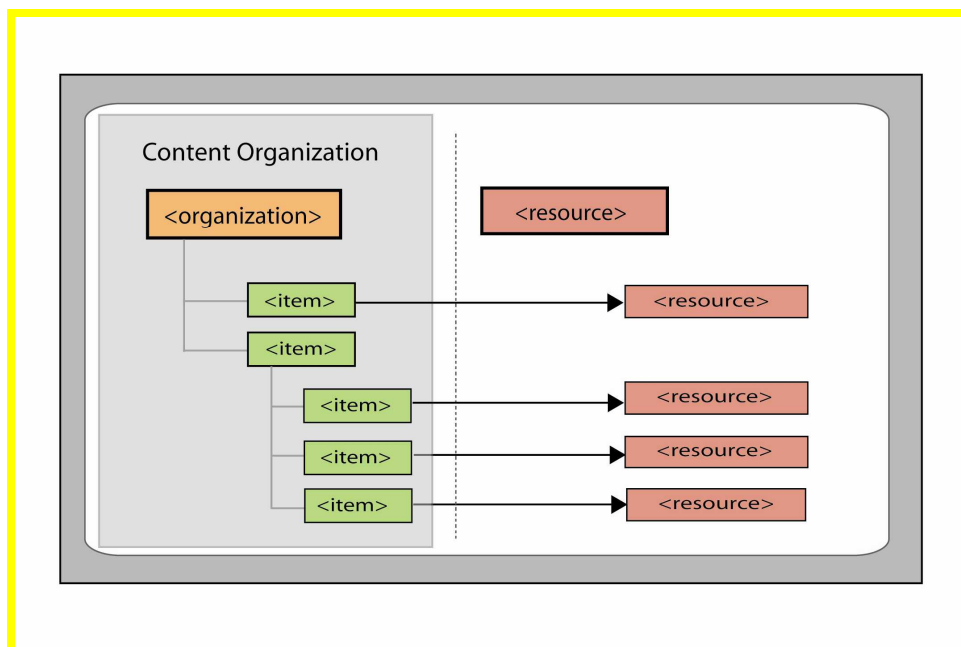


図 2.1.4a コンテンツオーガニゼーションの概念的表現

コンテンツオーガニゼーションはメタデータで記述することができ、これによって再利用の機会を設け、メンテナンスを容易にすることができる。

シーケンシングはアクティビティにだけ適用される。アクティビティのシーケンシングの意図はコンテンツオーガニゼーションの中で、アクティビティを他と関連付け、各アクティビティにシーケンシング情報を関連付けることで定義される。LMS は、コンテンツオーガニゼーションの中に記述されたシーケンシング情報を解釈し、学習時の学習リソースの実際の順序を制御するためにシーケンシング動作を適用しなくてはならない。

この開発方法は、独立したコンピュータベースドトレーニング（CBT）オーサリングツールを用いた教材開発方法からの脱却を意味する。過去において、これらのツールは、典型的に、学習者

が次にコースのどの部分を見るのかを制御する全てのシーケンシングおよびナビゲーション情報を、固有のデータフォーマットで埋め込んでいた。ほとんど全てのケースで、オーサリングツールおよびシステムは、コンテンツのために、独自でしばしば独特なシーケンシング手法を定義していた。SCORM の登場および相互運用可能な開発戦略への移行以前は、異なった編集環境間でコンテンツを共有するのは非常に困難であった。また異なったシーケンシング要件を含む他のコンテキストでコンテンツを再利用することも同様に困難であった。

SCORM では、シーケンシング情報は、コンテンツオーガニゼーション中のアクティビティにおいて定義され、それらのアクティビティと関連付けられた学習リソースの外部にある。定義されたシーケンシング動作を適用して、アクティビティに関連付けられた学習リソースを起動するのは LMS の役割である。これは概念的に重要である。なぜなら、コースの文脈固有のシーケンシング情報が学習リソースに組み込まれていると学習リソースの再利用は制限されるからである。例えば、学習リソースに、特定の条件で他の学習リソースに直接結びつけられた分岐を含むと、この学習リソースは、2 番目の学習リソースが適用可能でも利用可能でない異なるコースでは利用できない。学習リソースの再利用の可能性は、それが独立していて自立可能なことに依存している。

しかしながら、SCORM は、いくつかの学習リソースが特定の学習タスクを達成する内部ロジックを含むことを認識している。そのような学習リソースは、自身の中で、学習者との相互作用によって分岐することがある。これらの分岐は全て自己完結したもので、スタンドアロンの学習リソースに関連し、LMS には見ることができない。重要なことは、内部分岐は、他のコンテンツオーガニゼーションで現れるか現れないかわからない外部の学習リソースを参照してはならないことである。これは、どの学習リソースが利用されどのように集約されるのかを決定するとき、コンテンツ開発者が注意すべき重要な領域である。

2.1.5. コンテンツアグリゲーション

コンテンツアグリゲーションは、活動と概念的な実体の記述の両方に用いることができる。コンテンツアグリゲーションは、機能的に関連したコンテンツオブジェクトの集合を作成する活動ないしプロセスの記述に用いることができる。SCORM コンテンツモデルの観点からは、コンテンツアグリゲーションは、このプロセスないし活動の一部として生成された実体の記述としても用いられる。しばしば、この用語は概略的にコンテンツパッケージを指すものとして用いられる。すなわち、コンテンツアグリゲーションは、コンテンツおよび所定のコンテンツ構造を提供するのに用いられ、システム間でやりとりされ、さらにリポジトリに格納することができる。

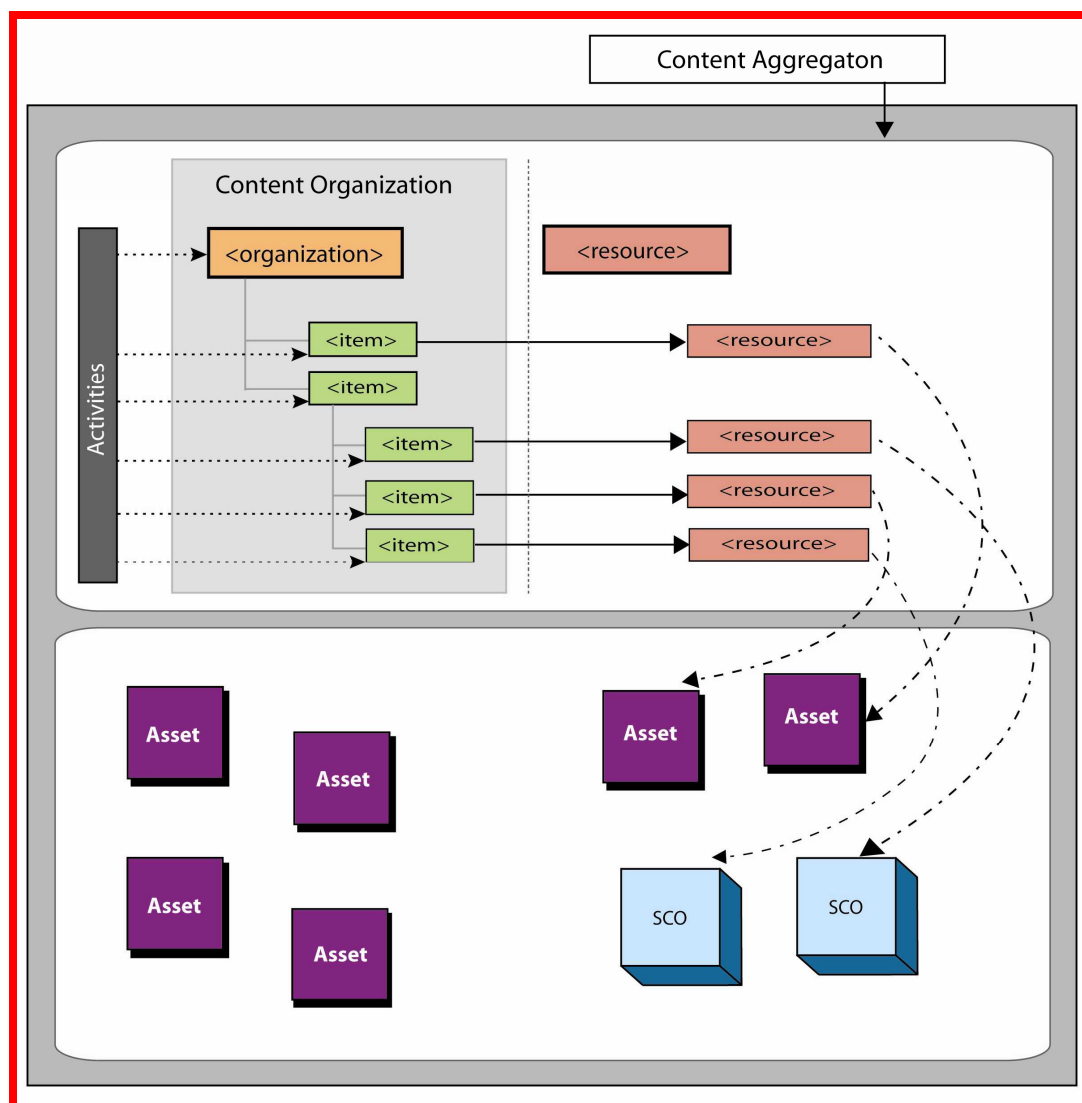


図2.1.5a コンテンツアグリゲーションの概念的表現

セクション 3

SCORM コンテンツパッケージング

このページは空白である。

3.1. コンテンツパッケージング概要

学習コンテンツが一旦設計され制作されると、コンテンツを、学習者、編集ツール、レポジトリもしくはLMSが利用可能とする必要がある。IMS コンテンツパッケージング仕様は、学習コンテンツを構築および交換する標準の手段を提供するように設計されている。

コンテンツパッケージの目的は、学習コンテンツを異なったシステム間もしくはツール間で交換する標準化された手段を提供することである。コンテンツパッケージは構造（もしくはオーガニゼーション）および学習コンテンツの集合の意図された動作を記述する場所を提供する。

コンテンツパッケージは、LMS 間や開発ツール間およびコンテンツレポジトリ間で学習コンテンツもしくは学習コンテンツの集合を動かすために用いられるように意図されている。IMS コンテンツパッケージング仕様 [3] はどんなシステムでもサポートできる共通の“input/output”フォーマットを提供する。

SCORM コンテンツパッケージングは IMS コンテンツパッケージング仕様の特定の要件とガイドラインのセット、すなわちアプリケーションプロファイルである。SCORM コンテンツパッケージングは IMS コンテンツパッケージング仕様を厳格に遵守し、アセット、SCO およびコンテンツオーガニゼーションをパッケージ化する追加の明確な要件および実装ガイダンスを提供する。

このセクションは以下のように構成される：

「セクション3.2: コンテンツパッケージ要素」は、コンテンツパッケージを取り扱う重要なコンセプトを定義する。これらの重要なコンセプトは、特定の要件を記述する以前に、コンテンツパッケージの基礎的な理解を得るのに役に立つ。

「セクション3.3: マニフェストの要素」はコンテンツパッケージマニフェストの構成を記述する。マニフェストは、コンテンツパッケージに対するパッケージング明細としての役割を果たし、コンテンツパッケージの要素を記述する。

「セクション3.4: コンテンツパッケージの構築」は、コンテンツパッケージ作成のプロセスを定義する。このセクションはコンテンツパッケージおよびマニフェストファイルの作成に焦点をあて、マニフェストのXML 部品およびこれらのXML 部品を利用する要件を記述する。

「セクション3.5: SCORM コンテンツパッケージアプリケーションプロファイル」は、特に、アセット、SCO およびコンテンツオーガニゼーションを含む SCORM 対応パッケージをどのように作成するかを定義する。このセクションは2種類のアプリケーションプロファイルとそのプロファイルに関連付けられた要件を記述する。

「セクション3.6: ベストプラクティスおよび実践的ガイドライン」は、コンテンツパッケージを作成もしくは処理するときのベストプラクティスおよびガイドラインを定義する。

3.2. コンテンツパッケージ要素

このセクションはコンテンツパッケージの概要やコンテンツパッケージを記述するために使用される専門的な名称およびコンテンツパッケージの制作を含んでいる。IMS コンテンツパッケージング仕様は、編集ツール、LMS およびランタイム環境でインターネットベースのコンテンツの相互運用可能性を提供するために利用されるデータ構成を記述する。IMS コンテンツパッケージング仕様の目的は、コンテンツを交換するのに利用される標準化された構造のセットを定義することである。IMS コンテンツパッケージング仕様の対象範囲は、コンテンツパッケージをインポート、エクスポート、統合および分解しようとするシステム間の相互運用可能性を定義することに焦点が当てられている。

コンテンツパッケージは2つの主要な部品を持つ：

- マニフェストファイル(imsmanifest.xml)と呼ばれるコンテンツ構造およびパッケージの関連するリソースを記述する特殊な XML ドキュメント。マニフェストに関する詳細は「セクション3.3: マニフェストの部品」参照。マニフェストはコンテンツパッケージのルートになくても構わない。
- コンテンツパッケージを構成する物理ファイル。

図 3.2a はコンテンツパッケージの部品を示す概念図である。

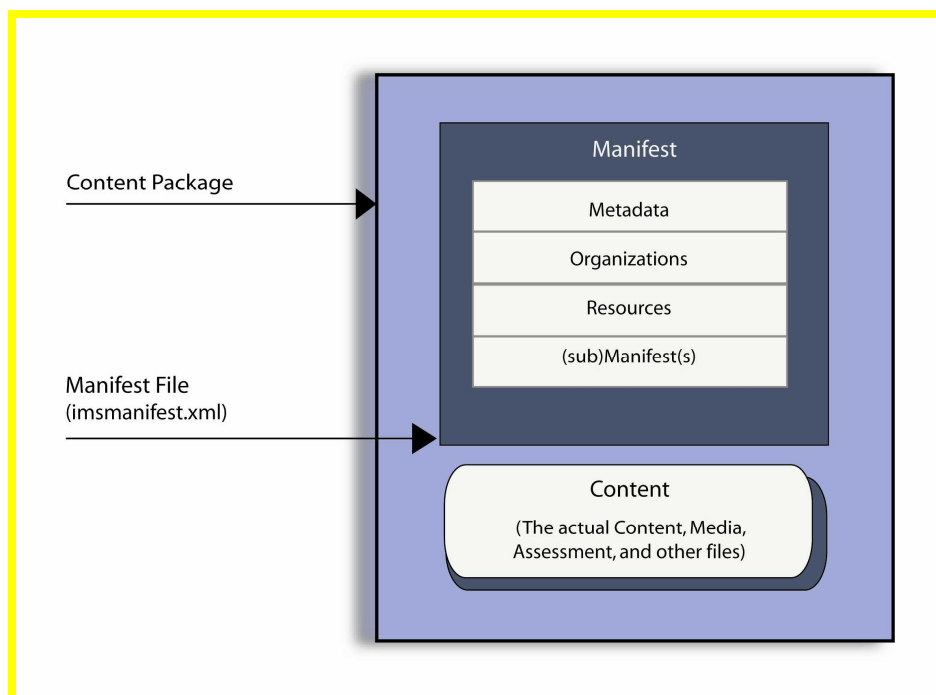


図 3.2a: コンテンツパッケージ概念図

3.2.1. パッケージ

パッケージは、学習の単位を表す。この学習の単位はコースの1部であることもあり、コースオーガニゼーションから外れても学習意義があり、コースの1部、コース全体もしくはコースの集合として独自に配信することもできる。パッケージが終点に到着すると、パッケージは分解もしくは統合されなければならない。パッケージはスタンドアロンになることができなければならない。つまり、分解されたときに、学習のためにパッケージ化されたコンテンツを利用するのに必要な全ての情報を含まなければならないことを意味する。

3.2.2. マニフェスト

マニフェストは、パッケージのコンテンツの構造化されたインベントリを含むXMLドキュメントである。コンテンツパッケージがエンドユーザに配信されることが意図されるならば、マニフェストもまた、どのようにコンテンツが組織されるかについての情報を含まなければならない。

マニフェストの対象範囲は柔軟性がある。マニフェストは、コースの文脈外でスタンドアロンでも成り立つコースの1部、コース全体、コースの集合、もしくはシステム間で運搬されるコンテンツの集合だけを記述することがある。コースの集合をパッケージ化するとき、コンテンツパッケージは、一般的には、学習者に配信するために、実際のLMSのランタイムシステムで分解されなければならない。この分解をどのように行うかは、SCORMのこのバージョンでは、対象範囲外である。現時点では、学習者に配信するコンテンツを提示もしくは保存するために、異なったLMSおよびレポジトリが、異なったメソッドを利用するので、非常に大きなもしくは非常に複雑なパッケージを、実際のLMSでどのように出すかについての統一見解もしくは標準はない。

一般的なルールは、パッケージは、常に一つのトップレベルマニフェストを含み、このマニフェストは一つもしくはそれ以上の（サブ）マニフェストを含む可能性があることである。トップレベルマニフェストは常にパッケージを記述する。組み込まれた（サブ）マニフェストは、それが対象とするレベル（コース、教育オブジェクトなど）でコンテンツを記述する。

ADL Note: IMS Global Consortium, Inc.,はIMSコンテンツパッケージ仕様の新しいバージョンの開発を行っている。IMSが解決しようとしている大きな課題のひとつは、（サブ）マニフェストに関して、その使用法、使用上の要求、XMLシンタックスの要求などである。現時点で、IMSの作業が終わるまで、ADLは（サブ）マニフェストを使用しないよう推奨する。（サブ）マニフェストに関する質問、懸案事項、新たな推奨項目はADLに送られたい。

マニフェストは以下の要件（IMSコンテンツパッケージで定義された通り）を遵守する：

- マニフェストファイルはimsmanifest.xmlというファイル名とする
- imsmanifest.xmlおよびそのサポーターコントロールファイル(DTD, XSD)のどれもコンテンツパッケージのルートに置く。オーガニゼーションの定義された機能を記述するために拡張が利用され、それらの機能がXMLで表現されたら、全てのコントロールファイルはパッケージのルートになくはならない。これはXMLインスタンスを有効にするために必要な、<adlcp:location>要素で参照されるものを含む全てのコントロールファイルを含む。
- IMSコンテンツパッケージングXMLバインディング仕様で定義された全ての要件、「セクション3.4: コンテンツパッケージの構築」で定義されたIMSバインディングへの全ての制約および追加要件

3.2.3. パッケージ交換ファイル (PIF)

パッケージ交換ファイル(PIF: Package Interchange File)は、圧縮されたアーカイブファイルの形でコンテンツパッケージ要素のバインディングである。PIF は `imsmanifest.xml`、全てのコントロールファイルおよびコンテンツパッケージで参照されたリソースを含む。SCORM はコンテンツパッケージは PIF として作成されることを推奨する。PIF は、システム間でコンテンツパッケージを運ぶのに利用される簡単な Web 配信フォーマットを提供する。PIF がコンテンツパッケージを提示するのに利用されると、SCORM は、PIF が RFC 1951 [12]に準拠することを要求する。この要求に加えて、SCORM は、アーカイブフォーマットは PKZip v2.04g (.zip)であることを要求する。この.zip ファイルは RFC1951 に準拠する。

3.3. マニフェストの要素

マニフェストファイルは、パッケージのコンテンツを記述するのに必要な情報を表現する。図 3.3.a はマニフェストの構成を示す。



図3.3a: マニフェストの部品

マニフェストは4つの主要セクションから成り立つ：

- メタデータ： コンテンツパッケージ全体を記述するデータ
- オーガニゼーション： コンテンツ構成もしくはスタンドアロンの単位もしくは教材の単位を構成する学習リソースのオーガニゼーションを含む
- リソース： コンテンツパッケージでバンドルされた学習リソースを定義する
- (サブ)マニフェスト： 論理的に組み込まれた単位の教材を記述する（スタンドアロンの単位として扱われることも可能）

3.3.1. メタデータ

メタデータは、データに関するデータと定義される。図 3.3a に示されたメタデータは、コンテンツパッケージ全体（コンテンツアグリゲーション）を記述するために利用される。このメタデータはコンテンツパッケージの検索や発見を可能にし、またコンテンツパッケージの特性を記述するためのメカニズムを可能にする。

ADL Note: メタデータは、マニフェスト内の様々なロケーションで、コンテンツパッケージの異なった観点を記述するのに利用されることができる（詳細はセクション 4.5.1: *Associating Meta-data with SCORM Components* 参照）。

3.3.2. オーガニゼーション

オーガニゼーション要素は、コンテンツがどのようにコンテンツパッケージで組織されるかを記述するのに使われる。オーガニゼーション要素は、一つ以上のオーガニゼーション要素を含み、各々の部品はパッケージのコンテンツに対するある特定の構造を記述する。IMS コンテンツパッケージ仕様の現在のバージョンは、コンテンツオーガニゼーションの一形態を定義するだけであり、これはツリーもしくは階層構造の形をしている。IMS コンテンツパッケージ仕様も SCORM も、この階層が特定の学習分類もしくは専門名称を表すべきかどうかを定義しない。言い換えれば、SCORM は、モジュール、レッスンそしてコンテンツオーガニゼーションで階層のレベルを記述するような用語を定義しない。そのような用語は、ある特定の方法で決定されがちである。従って、専門名称の選択は、コンテンツ開発者に任されている。

3.3.2.1. 複数のコンテンツオーガニゼーション

IMS コンテンツパッケージ仕様は学習リソースをこれらの学習リソースが組織される方法から切り離し、異なった文脈もしくは利用方法の中で、同一の学習リソースが、一回以上利用されることを可能にする。

LMS で学習者にコンテンツパッケージを配信するために、少なくとも一つのコンテンツオーガニゼーションがなければならない。いくつかのケースでは、コンテンツをパッケージで利用する異なった方法を定義するのは有効である。各々の方法は、異なったコンテンツオーガニゼーションで表現される。いくつかのコンテンツオーガニゼーションがあったとして、パッケージを利用する実体がどのコンテンツオーガニゼーションを利用するか決定できないときには、デフォルトで利用されるように指定されたオーガニゼーションが常にある。

3.3.2.2. コンテンツオーガニゼーション

コンテンツオーガニゼーションは、コンテンツパッケージのコンテンツが利用に当たってどのように組織されているかを記述する。SCORM 1.2 で、コンテンツオーガニゼーションに対して唯一定義された利用法は、コンテンツパッケージにおけるリソースに対する目次としてである。このバージョンの SCORM では、コンテンツオーガニゼーションは学習リソースを活用する学習アクティビティの階層を記述する方法としても利用される。これらのリソースはマニフェストの他の部分で記述される。

コンテンツオーガニゼーションは、コンテンツパッケージの物理的な構造もしくはマニフェストの構造と間違えられるべきではない。例えば、コンテンツパッケージのファイルは、しばしば、フォルダの階層構造で組織化されるが、この構造自体はコンテンツパッケージの利用者にパッケージのコンテンツをどのように利用するかを教えることはできない。

コンテンツオーガニゼーションの目的は、学習リソースの集合を利用する教材の意味のある単位を特定する手段を、コンテンツ開発者に提供することである。そのような教材の単位は学習アクティビティの階層構造であり、この階層構造に対して固有の動作およびルールが記述され、このアクティビティ構造と関連する動作がどんな SCORM 対応 LMS 環境でも再生可能となる。

コンテンツオーガニゼーションで定義されたアクティビティに対し、コンテンツ開発者は、特定の動作やルールを定義することもしないこともある。特定のルールがなければ、コンテンツオーガニゼーションは、コンテンツパッケージで定義された学習リソースを通して自由にナビゲートするのに利用される単なるマップである。特定のルールや動作を追加することにより、コンテンツオーガニゼーションは、LMS が学習者経験および学習リソースの利用をどのように管理しなければならないかを定義するガイドとなる。

コンテンツオーガニゼーションは、学習リソースを利用する学習アクティビティの階層を通じて、学習者を導くための構造化された学習リソースのマップもしくは構造化されたアクティビティマップとして見ることができる。あるコンテンツ開発者は、コンテンツオーガニゼーションを学習リソースの目次として構築するよう選択できるし、他のコンテンツ開発者は、必要なときだけ学習リソースを呼び出し、学習経験を通して道先案内としてコンテンツオーガニゼーションを構築するよう選択できる。別のコンテンツ開発者は、発見アクティビティは、いくつかの学習リソースの自由な利用を含む一方で、他のアクティビティはより形式的に管理されるようにコンテンツオーガニゼーションを構築できる。

LMS は、コンテンツパッケージおよびそのコンテンツの独自の内部表現を作成できる。SCORM でのコンテンツオーガニゼーションの仕様は、LMS システムが、コンテンツオーガニゼーションモデルを採用する必要があること、もしくは同じ構造のオーガニゼーションを利用してパッケージ部品を保存する必要があることを意味しない。

IMS コンテンツパッケージング仕様で定義されたオーガニゼーション要素は、コンテンツ構造を表すための情報の枠組みを提供する。設計により、オーガニゼーション要素は、オーガニゼーションの全てのアイテムに対して、メタデータ、動作ルールといった追加情報や指示を追加する機能を持つ。SCORM は、そのようなオーガニゼーション要素を「コンテンツオーガニゼーション」と呼ぶ。

コンテンツオーガニゼーションは、実際の学習リソースの集まりではなく、これらのリソースを記述していない。IMS コンテンツパッケージングモデルは学習リソースを搭載する方法を示し、配信するのに要求される物理的なファイルの全てへのリンクを含み、パッケージ内で物理ファイルとして含まれない外部参照リソースを含む一つ以上の学習リソースに属するファイル間の関係を識別する方法を提供する。

3.3.2.2.1. コンテンツとコンテンツ集合の編集

コンテンツオーガニゼーションは、学習リソースの集合の構造を表す手段を提供する。これは、学習コンテンツの設計に関する比較的新しいアプローチである。過去には、CBT 編集ツールが、コースの部分およびこれらの部分がいつどのように学習者に提示されるかの手段を提供していた。学習リソースおよびコンテンツオーガニゼーションは通常切り離すことができず、同一のツールおよび固有のデータフォーマットを利用して開発された。インターネットベースのテクノロジーおよび再利用可能なコンテンツオブジェクト作成の考えへの移行が、編集プロセスを大幅に変更した。

SCORM では、コンテンツオーガニゼーションに組み込まれた指示に従ってコンテンツを配信するのは、LMS の役割である。これは、コンテンツパッケージに含まれた学習リソースがいつどのように学習者に提示されると設計者が意図したかを、LMS が知っていなければならないことを意味する。パッケージマニフェストのオーガニゼーションセクションに存在するコンテンツオーガニゼーションは、設計者が LMS にこのような情報を提供することを可能にする。これは、ある単位の教材の編集は、おそらく完全に異なった編集ツールを利用して、学習リソースを編集または収集することと、それらの学習リソースを利用するコンテンツオーガニゼーションを編集することのふたつから成り立っているということを意味する。

SCORM では、編集により 2 つの異なった成果物が得られる：LMS が指示として利用し、ランタイムに処理される、編集されたコンテンツオーガニゼーション情報、および、コンテンツオーガニゼーションの指示によりブラウザ環境で起動される、編集された学習リソースである。多くの古い CBT モデルと反して、ここでは、構造はコンテンツから切り離されている。構造は、今では完全に可視化され標準的な形で表現されるようになっている。コンテンツパッケージは、異なった LMS 環境で等価な学習経験を生み出すために、予測可能な方法で配信される。

3.3.2.3. コンテンツ構成の表現

SCORM コンテンツオーガニゼーションは、コンテンツ構造の異なった観点を定義するように意図された要素を含む：

- **コンテンツ階層：**これは、学習リソースもしくは学習リソースを利用するアクティビティの論理的な構成を表す目次のようなツリー構造の表現である。すべてではないが、多くのケースでは、この階層的なツリーは、学習者が教材を学習するように作成者が意図したデフォルトの順序に対応した定められた順序で検索される
- **メタデータ：**これは、コンテンツオーガニゼーションで定義されたアクティビティに関するオプションの文脈依存の記述的なデータである。そのようなメタデータは、ある特定の学習リソースがある特定のコンテンツオーガニゼーションどのようにで利用されるかを記述するのに利用される（学習リソースが特定のアクティビティで利用されるときに学習者にみあうコンピテンシや学習目標など）
- **シーケンシング、適応型シーケンシングおよびナビゲーション：**学習者がコンテンツを閲覧する際、どの学習リソースが学習者に提示されるかをコンテンツ開発者がコントロールしたい場合、コンテンツオーガニゼーションにオプションの指示が組み込まれることがある。デフォルトではシーケンシングおよびナビゲーション指示が定義されないと、学習者は自由にどのコンテンツアイテムを選んでも良い。特定の指示を加えることで、このデフォルト動作を修正することができる。例えば、コンテンツオーガニゼーションのアイテムにフロー指示を追加することは、オーガニゼーションツリーで定義された順序でナビゲーションをガイドするよう LMS に指示を出すことになる。より複雑な適応型シーケンシングは、ある特定の学習リソースの完了ステータス、もしくは、より複雑なユーザの好みもしくはアセスメント結果の計算に基づいている。

コンテンツオーガニゼーションはコンテンツの集合に様々なアプローチを表現するように意図されている。コンテンツオーガニゼーションは、単純な数行の HTML (Hypertext Markup Language) もしくは短いメディアクリップのような、非常に小さい学習リソースから、LMS に追跡される高度な双方向の学習リソースまでの広い範囲のコンテンツ集合を表す。SCORM コンテンツアグリゲーションモデルは、コンテンツの複雑性、学習単位の階層レベルの数（分類等）、取り入れられている学習手法などに中立的である。

表 3.3.2.2a は、SCORM コンテンツオーガニゼーションで表現可能な、いくつかのアメリカとカナダの軍機関で利用されたカリキュラム区分モデルのいくつかの例を示す

表 3.3.2.2a: カリキュラム区分モデルの例

米国陸軍	米国空軍	米国海兵隊	カナダ軍
Course	Course	Course	Course
Module	Block	Phase	Performance Objective
Lesson	Module	SubCourse (Annex)	Enabling Objective
Learning Objective	Lesson	Lesson	Teaching Point
Learning Step	Learning Objective	Task	
		Learning Objective	
		Learning Step	

3.3.2.3.1. コンテンツ階層

学習リソースの集合は、通常、コンテンツオーガニゼーションとして表現できる階層に構成される。学習リソースを構成するもう一つのアプローチは、特定の学習経験を提供するためのリソースに依存するアクティビティの階層を定義することである。このケースでは、階層のトップレベルがメインアクティビティである。メインアクティビティはサブアクティビティから成り立つことがあり、これらのサブアクティビティもまたそのサブアクティビティから成り立つことがある。設計手法により、この階層構造のグループ分けが、学習者へ配信するためにどのようにコンテンツが構成されるかを表す、コース、チャプター、トピック、あるいは、同様の言葉の概念を表すために利用される。

IMS コンテンツパッケージング仕様は、コンテンツ階層を表すのに利用する用語を定義している。図 3.3.2.2.1a は、コンテンツオーガニゼーションがどのように学習リソースの集合にマッピングされる階層構造を形成するかを示す（IMS コンテンツパッケージング仕様によって定義されているように）。階層構造は、入れ子になったアイテム要素のツリーである。ツリーのルートはオーガニゼーション要素である。アイテム要素は、アイテムがアクティビティとして利用されるときに起動される特定の学習リソースを参照する。同一のリソース要素が、一つ以上のアイテム要素によって参照されることもありうる。

SCORM および IMS シンプルシーケンシング仕様[5]は、IMS コンテンツパッケージング仕様のアプリケーションプロファイルであり、二つの制約が加わる。一つは、リーフアイテム（子を持たないアイテム）だけが学習リソースを参照することができるということである。もう一つはオーガニゼーションは最低一つのアイテムを含まなければならないということである。

さらに、SCORM は、学習リソースは以下の二つのうちどちらか一つのタイプであることを要求する。SCO もしくはアセットのどちらかである。これらは、リソースのセクションで詳細に説明されている。

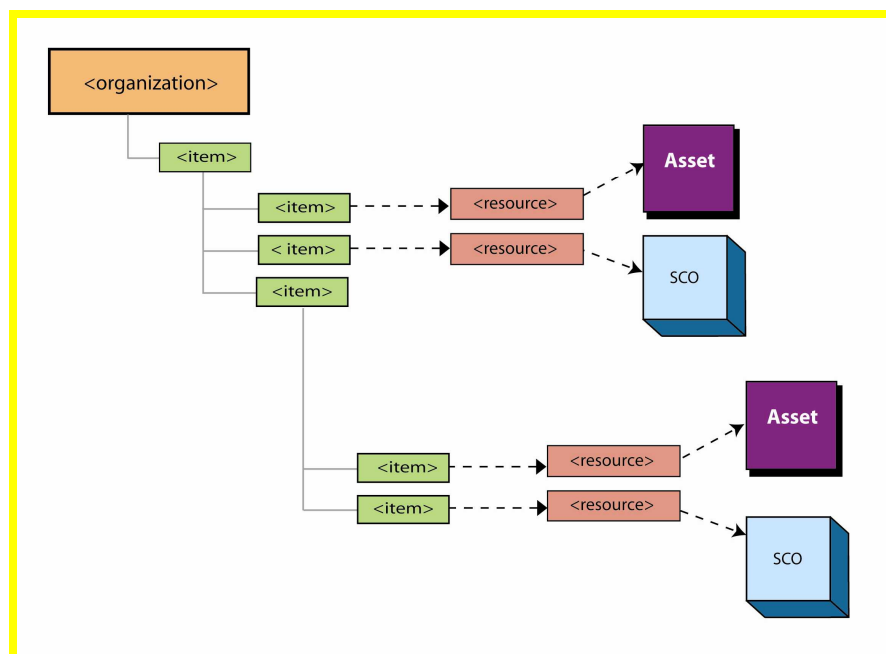


図 3.3.2.3.1a: IMS コンテンツ階層用語

3.3.2.3.2. メタデータ

学習リソースが再利用可能となることを意図して作成されるとき、メタデータで学習リソースを記述するのがベストプラクティスである。メタデータは、コンテンツパッケージもしくはレポジトリに保存された学習リソースが発見されるようにする。リソースを実際に関いて検査することなく、メタデータを検査するだけで再利用することを決定できるので、学習リソースはより一層再利用できるようになる。そのようなメタデータは、学習リソースがどこで発見されるかに関わらず、特定の学習戦略の文脈での特定の利用からは独立して、学習リソースを記述するので、文脈依存でないと考えられる。例えば、針の穴を通すことを教えるという単純な SCO を考えてみよう。SCO を記述するメタデータは、必要とされるスキル（糸を針の穴に挿入する）を記述し、さらにシミュレーションが学習経験の 1 部であることを記述する。しかし、このメタデータは、針がどのように利用されるかを記述しない。学習リソースの利用が、学習戦略の一部として定義されると、追加のメタデータがそのリソースを利用するアクティビティを記述するのに利用されることがある。

ある特定の学習戦略に固有なメタデータは文脈依存のメタデータと呼ばれ、コンテンツオーガニゼーションに組み込まれている。例えば、メタデータはある特定のアクティビティが、何故学習シーケンスの中にある特定の場所に挿入されているかの説明を含んでいるのである。

文脈に依存しないメタデータは、通常、不変のメタデータを参照し、デジタルアセット、コンテンツオブジェクトもしくはオブジェクトの集合を記述する。反対に、文脈に依存するメタデータは、通常、ある特定のコンテンツオーガニゼーションのコンテキストでしか意味をなさないメタデータを表わす。

メタデータを、学習リソースおよび学習リソースの集合に対して開発し適用することは、多くの人にとって新しいコンセプトである。これを実行するためのベストプラクティスは、今後開発されていく。いくつかのケースでは、メタデータに対する主な目的は、コンテンツの発見と再利用である。他のケースでは、主な目的は完全に情報提供の目的であり、コンテンツオーガニゼーションの中で記述された学習オブジェクトもしくはアイテムの設計と意図に関する情報を編集者に提供することである。メタデータは学習者がコンテンツをナビゲートするのを助けるために提供することができるという説もある。メタデータの共通利用に関する共通認識は、今のところできていないが、メタデータの様々な有効な利用法に対して、これらの仕様で規定ができている。

コンテンツパッケージが、学習者への配信のためだけに意図されていて、要素を他のオーガニゼーションで再利用するために、コンテンツパッケージを再組織化、もしくは分解する意図がなければ、メタデータによって、コンテンツパッケージおよびそのマニフェストを保存、送信、管理するのがより高価になるだけなので、コンテンツパッケージの各要素に対して詳細なメタデータを追加するのは、生産性が低くなる。その一方で、将来、再利用に向けて、コンテンツパッケージを修正、再組織もしくは分解する可能性が少しでもあれば、再利用される可能性のある各要素に対してメタデータを提供するのは道理にかなっている。低コストの配信パッケージが要求された場合は、メタデータが除かれることがあるにしても、コンテンツの過去の経験から、可能性がある場合はメタデータを追加するのがおそらくベストであると言える。

3.3.2.3.3. シーケンシング、適用性のあるシーケンシングおよびナビゲーション

シーケンシングおよびナビゲーションは、編集者もしくはコンテンツ開発者が意図したように特定の学習経験を提示するために、LMS が従わなければならない動作を指す。この学習経験は、学習者がコンテンツオーガニゼーションで、いかなるアイテムをどんな順序でも選べるという自由な動きかもしれないし、コンテンツオーガニゼーションの構造の流れにガイドされるのかもしれない。

ない。LMS が追跡可能な学習者のパフォーマンスや他の要因に依存する異なった動作によって、学習経験は適応的になることができる。

SCORM は、ある特定のコンテンツオーガニゼーションに対してシーケンシングおよびナビゲーションを整理するデフォルトのシーケンシング情報を定義する。しかし、デフォルトシーケンシング情報は、自由閲覧動作だけを提供する。多くのコンテンツ開発者および教材設計者は、特定の学習あるいは教授戦略を利用することを好む。それらのケースでは、コンテンツ開発者は、LMS がどのように学習経験を管理するかを指示する特定のシーケンシング情報を定義することができる。

シーケンシング情報は、コンテンツオーガニゼーションツリーの要素と関連付けられ、各要素は、シーケンシング情報のコントロールの下で学習者が取り組むアクティビティを表す。シーケンシング情報の適用は、一般的には学習リソースの起動か、コンテンツ開発者が定義した制約内で学習者が取らなければならない選択のどちらかに帰着する。シーケンシングおよびナビゲーション情報は、コンテンツオーガニゼーションの一部であり、そのコンテンツオーガニゼーション自身がパッケージマニフェストの一部であるので、パッケージが SCORM 対応のいかなる LMS でも同一の学習経験を配信するように、意図された動作はコンテンツパッケージに組み込まれる。

SCORM シーケンシングおよびナビゲーションは、他のことと並んで、高度に適応性のあるアクティビティシーケンシングを定義する能力を提供する。例えば、学習者が何らかのタスクを完了したかどうか、認められるスコアを得点したかどうか、特定の目標を達成したかどうか次第で、他のアクティビティに条件付で分岐させることができる。コンテンツオーガニゼーションに組み込まれたシーケンシングおよびナビゲーション情報は、同一のパッケージの他の学習リソースが過去のアクティビティでどのように利用されたかに基づいて、LMS が学習者にコンテンツパッケージ内の学習リソースをどのように利用させるかを指示できる。

過去には、CBT 編集ツールは、一般的に、固有のデータフォーマットでコーディングされた作り込みのシーケンシングおよびナビゲーション機能を提供した。しかしブラウザベースのコンテンツを異なった LMS システムを通して運用するといった新しい要求が登場した。もう一つの要求は、構造と学習戦略ロジックを、戦略を実装される学習リソースから切り離すことを可能にすることであった。これは、異なった LMS 環境で、コンテンツオーガニゼーションが移動でき再利用されるように、適応性のあるシーケンシングおよびナビゲーション動作を定義し、コーディングする何らかの方法を標準化するニーズに結び付いた。

特定のタスクを効果的に訓練したり、複雑な役割や責任に対して学習者に準備させるために要求される様々な複雑な設計アプローチのために、シーケンシングおよびナビゲーションの標準化プロセスは困難であった。SCORM の過去のバージョンは、純粹に自由に学習することを可能にするだけで、特定のシーケンシング機能を提供していない。なぜなら、それは困難で複雑な課題であり、役に立つ解決策を打ち出すにはもっと時間を要したからである。学習設計のコミュニティでは、多くの、しばしば異なる要求がある。全員を満足させるアプローチは見つかっていないが、SCORM で利用される IMS SS 仕様[5]に基づいたアプローチは、様々な学習設計アプローチを可能にする柔軟性が充分ある。

「セクション 5: SCORM シーケンシングおよびナビゲーション」は、シーケンシングおよびナビゲーションルールが、IMS SS 仕様および SCORM に準拠して、どのようにパッケージマニフェストのコンテンツオーガニゼーションの XML 表現に組み込まれているかを説明している。この仕様は、コンテンツオーガニゼーションスキームをシーケンシング指示で拡張し、シーケンシングおよびナビゲーション情報がコンテンツパッケージに強固に関連付けられるようにする。これらの指示は、ルール、イベント、条件を表現するための共通モデル、および、様々なシーケンシングおよびナビゲーションメソッドと関連付けられたランタイム動作に基づいている。

IMS SS 仕様は、システムが、学習者の学習リソースとの相互作用に一貫性のある対応をしながら、学習リソースを予想可能な形で配信することを可能にする。意図されたアプローチは、コンテンツ開発者にシーケンシングおよびナビゲーション動作もしくは学習戦略を学習リソース自体と独立に定義させることによって、学習リソースの再利用の可能性を促進する。適応性のあるシーケンシング情報はコンテンツオーガニゼーションでコーディングされ、学習リソースが複数コンテキスト（各々が各自のシーケンシングおよびナビゲーション情報を持つ複数の異なったマニフェストもしくはオーガニゼーション）で再利用されることを可能にする。

3.3.3. リソース

マニフェストのリソース要素は、パッケージ中の物理的なファイルおよび外部リソースを記述する。これらのファイルはメディアファイル、テキストファイル、アセスメントオブジェクトもしくは他の電子媒体のデータである。ファイル間の概念的グループ分けおよび関係はリソース要素内で表現することができる。リソースの組み合わせは、一般的には「コンテンツ」として分類される。リソースは、リソース相互の構造を提供するオーガニゼーション要素内の様々な箇所から参照される。

図 3.3.3a では、単一のリソースが複数の部品から成り立っている。SCORM では、これらの部品は、単純なるアセットである。リソースが LMS と交信するように作られていれば、(SCORM ランタイム環境ブック[2]参照)、そのリソースは SCO である。リソースが LMS と交信するように作られていなければ、リソースはアセットとみなされる。リソース要素の集合は、オーガニゼーションが参照できるリソースを構成する。このリソースの集合とオーガニゼーションが、コンテンツオーガニゼーションを定義する。

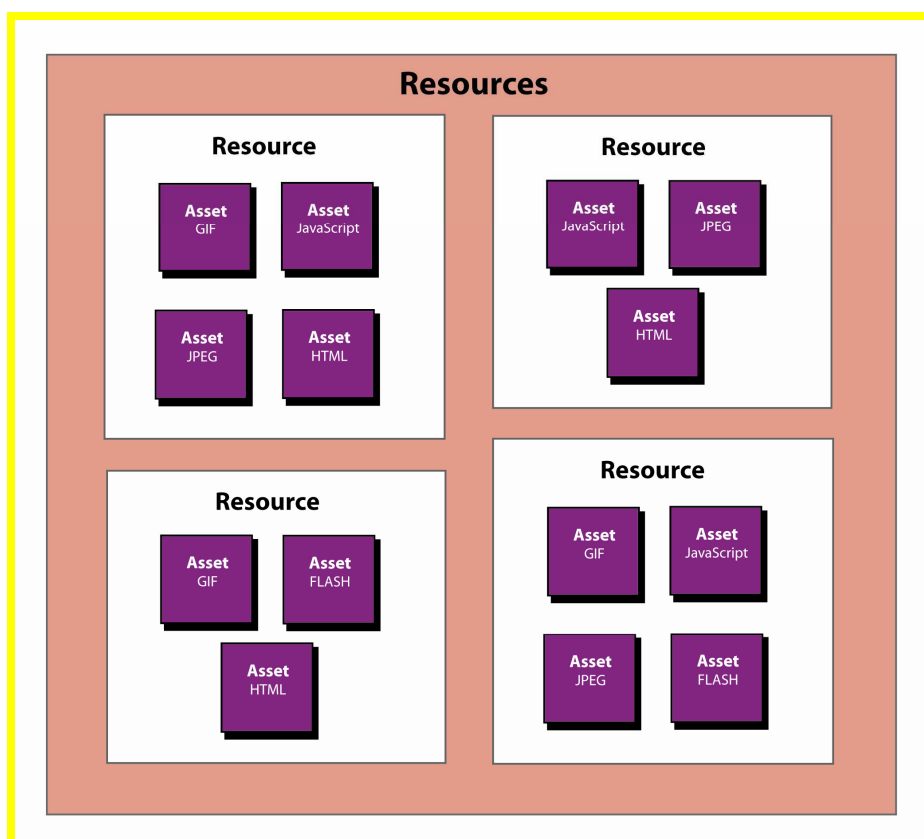


図3.3.3a: マニフェストリソース

リソースはリソース全体の物理構成を記述する。リソースの要素はリソース内のファイルとして列挙される。

3.3.4. コンテンツ

コンテンツ（すなわち物理ファイル）要素は、リソース要素で参照される実際のファイルを表す。これらのファイルはコンテンツパッケージに実際に含まれるローカルファイルや、URI で参照される外部ファイルである。コンテンツパッケージに含まれるすべての物理ファイルは、コンテンツパッケージを交換する際、マニフェストで宣言され、参照されていなくてはならない。マニフェストで参照されないファイルをコンテンツパッケージに含めることは、コンテンツパッケージをインポート、実行、エクスポートする際に、広範囲な問題を引き起こす可能性がある。これらのプロセスのいずれかにおいて、物理ファイルはコンテンツパッケージを完結させるために必要となる。システムは、コンテンツパッケージの構成を決定するためにマニフェストを使用する。物理ファイルがマニフェストに挙げられていない場合、あとでコンテンツパッケージが破壊される可能性がある。

3.4. コンテンツパッケージの作成

このセクションは SCORM コンテンツパッケージを作成する際の要件を提示する。このセクションは、SCORM に適用された IMS コンテンツパッケージング仕様の XML バインディングを説明する。この XML バインディングの作成をガイドするいくつかの特定のルールがある：

- XML バインディングは、W3C の XML 1.0 仕様[6]に準拠する
- XML バインディングは、IMS コンテンツパッケージング情報モデルの定義された構造を維持しなければならない

いくつかの要件は他の様々な仕様や規格から取り込まれてもいる。要件の多くは、IMS コンテンツパッケージング仕様で定義された要件から受け継がれている。他のいくつかの仕様および規格は、XML および他のインターネット技術の性質に基づいて、暗黙のうちに受け継がれている。

このセクションは、また、SCORM コンテンツパッケージアプリケーションプロファイルの各々の要件を定義する：

- リソースパッケージアプリケーションプロファイル： 学習リソース（SCO およびアセット）の定義されたオーガニゼーション無しに、学習リソースの集合を作成するためのコンテンツパッケージ。これらの学習リソースはお互いに関係を持つ必要がない
- コンテンツアグリゲーションパッケージアプリケーションプロファイル： 学習リソースおよびその意図された静的構成およびシーケンシング要件を作成するコンテンツパッケージ（マニフェストは一つ以上の学習リソースの組織を含む）

詳細はセクション 3.5: *SCORM コンテンツパッケージアプリケーションプロファイル Content Package Application Profiles* 参照。

3.4.1. マニフェストファイル

以下のセクションは、imsmanifest.xml ファイルを作成するための要件を定義する。マニフェストはパッケージのコンテンツの構造化されたインベントリである。パッケージが、ユーザに配信されるよう意図される場合、マニフェストは、どのようにコンテンツが組織されるかに関する情報を持つ。imsmanifest.xml は、名称が示すように、XML ファイルである。このセクションは、IMS コンテンツパッケージング仕様で定義された各要素に対する要件を定義する。

いくつかの要素は、繰り返しとデータタイプの一方ないし両方を記述するのに、用語 SPM (smallest permitted maximum) を利用する。SPM は、コンテンツパッケージを処理するアプリケーションが、最低その数の要素もしくはその長さの文字列を処理すべきであるが、それ以上の数をサポートすることは自由であることを示す。

要素のデータタイプおよびフォーマットは、XML Schema Part 2: Datatypes W3C Recommendation [13]に指示されたデータタイプによって定義される。XML 要素の順序は、マニフェストの IMS Content Packaging XML Binding によって定義されている通りである。

下記の表は SCORM コンテンツパッケージングアプリケーションプロファイル（セクション 3.5 参照）要件を記述するのに利用される：

表3.4.1a: SCORM コンテンツパッケージングアプリケーションプロファイルテーブルフォーマット

SCORM コンテンツパッケージングアプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	<requirement>
リソース	<requirement>

左の欄は、SCORM コンテンツパッケージングアプリケーションプロファイルというタイトルになっている。この欄は、SCORM で定義されたアプリケーションプロファイルのタイプを記述する。右の欄は、マニフェスト繰り返し要件というタイトルになっている。この欄は imsmanifest.xml ファイルの XML 要素に対する繰り返し要件を定義する。

表3.4.1b: 繰り返し要件の説明

繰り返し要件	説明
1 and only 1	この要素は親要素内で 1 回だけ存在しなければならない
0 or More	この要素は親要素内で 0 回以上存在することが可能である
1 or More	この要素は親要素内で 1 回以上存在しなければならない
0	この要素は許されていない
0 or 1	この要素は親要素内で 0 回か 1 回存在することが可能である

表3.4.1b は、このセクションで利用される繰り返しのタイプを定義する。各タイプには簡単な説明が付いている。

3.4.1.1. <manifest> 要素

<manifest>要素は、メタデータ、オーガニゼーションおよびリソースリファレンスを包含する再利用可能な学習の単位を表す[3]。<manifest>要素は、imsmanifest.xml ファイルのルート要素ノードである。ルート<manifest>内の他の<manifest>要素は、集約、分解、再利用のために、ファイル、メタデータおよびオーガニゼーション構造を区分するために利用される。これらの子<manifest>要素は、(サブ) マニフェストと呼ばれる。(サブ) マニフェストは、セクション「3.4.2 (サブ) マニフェスト」で詳細に説明されている。

ADL Note: IMS Global Consortium, Inc., は IMS コンテンツパッケージ仕様の新しいバージョンの開発を行っている。IMS が解決しようとしている大きな課題のひとつは、(サブ) マニフェストに関して、その使用法、使用上の要求、XML シンタックスの要求などである。現時点で、IMS の作業が終わるまで、ADL は (サブ) マニフェストを使用しないよう推奨する。(サブ) マニフェストに関する質問、懸案事項、新たな推奨項目は ADL に送られたい。

全ての namespace 宣言は<manifest>要素内で宣言しなくてはならない。これは、IMS および ADL に対して拡張と考えられる namespace も含まれる。これは要求されるものではないが、XML 仕様に基づき、ADL はこれをベストプラクティスとみなし、ベンダやツールにこの情報を提供するよう勧めている

XML 名前空間: http://www.imsglobal.org/xsd/imsdcp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <manifest>

SCORM 要件: このマニフェストは IMS マニフェストのルート要素ノードである。ルート <manifest>要素は、一回だけ存在する

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	1 and only 1

データタイプ: <manifest>要素は親要素である。親要素は自身に関連する値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<manifest>要素は以下の要素/属性を含む:

属性:

- **identifier** (必須) - この属性はマニフェストを識別する。identifier はマニフェスト内で一意である[3]。identifier 属性は、一般的に、マニフェスト開発時に編集者もしくは編集ツールが提供する。XML Data Type: xs:ID.
- **version** (オプション) - version 属性はマニフェストのバージョンを識別する[3]。同一の識別子を持ったマニフェストを区別するのに利用される。値の SPM は 20 文字である。XML Data Type: xs:string.
- **xml:base** (オプション) - xml:base 属性はマニフェストに含まれるコンテンツファイルに対する相対的なパスのオフセットを表す[3]。この要素の利用は World Wide Web Consortium (W3C)で開発された XML Base [7]仕様で定義される。値の SPM は 2000 文字である。XML Data Type: xs:anyURI.

要素:

- <metadata>
- <organizations>
- <resources>
- <manifest>
- <imsss:sequencingCollection>

例:

```
<manifest identifier="SAMPLE1" version="1.3" xml:base="mycontent"
  xmlns="http://www.imsglobal.org/xsd/imsdp_v1p1"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsdp_v1p1
    imscp_v1p1.xsd
    http://www.adlnet.org/xsd/adlcp_v1p3
    adlcp_v1p3.xsd">
  <!-- imsmanifest contents -->
</manifest>
```

コード例3-1

3.4.1.2. <metadata>要素

<metadata>要素は、マニフェストを記述するメタデータを含み[3]、コンテンツパッケージ（コンテンツアグリゲーション）全体を記述する情報を含む。<metadata>要素はコンテンツパッケージで定義されたメタデータに対するルートノードとみなされる。これは、コンテンツパッケージに対する全てのメタデータは、<metadata>要素の子として定義されることを意味する。

XML 名前空間: http://www.imsglobal.org/xsd/imsdp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <metadata>

SCORM 要件: SCORM では、全ての<manifest>要素は、<metadata>要素に対する以下の繰り返し要件を含まなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	1 and only 1

データタイプ: <metadata>要素は親要素である。親要素は自身に関連する値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<metadata>要素は以下の要素/属性を含む:

属性:

- なし

要素:

- <schema>
- <schemaversion>
- {Meta-data}

例:

```
<manifest identifier="SAMPLE1" version="1.3" xml:base="mycontent/"
  xmlns="http://www.imsglobal.org/xsd/imsdp_v1p1"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsdp_v1p1
    imscp_v1p1.xsd
    http://www.adlnet.org/xsd/adlcp_v1p3
    adlcp_v1p3.xsd">

  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
    <adlcp:location>packageMetadata.xml</adlcp:location>
  </metadata>
</manifest>
```

コード例3-2

3.4.1.3. <schema> 要素

<schema>要素はマニフェストを定義しコントロールするスキーマを記述する [3]。この要素はパッケージを表すメタデータの子なので、この要素はマニフェストの要件をコントロールするスキーマを記述するのに利用される。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <schema>

SCORM 要件: SCORM では、<schema>要素は以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	1 and only 1

データタイプ: <schema>要素は文字列として表現される。XML Data Type: xs:string.

SCORM では、<schema>要素は以下の制約付きの語彙トークンを含むことを要求する：

- **ADL SCORM:** この制約付きのトークンは、コンテンツパッケージが SCORM で定義された要件に準拠して作られたことを示す

例：

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
  </metadata>
</manifest>
```

コード例3-3

3.4.1.4. <schemaversion>要素

<schemaversion>要素は上記のスキーマ (<schema>)のバージョンを記述する [3]。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <schemaversion>

SCORM 要件: SCORM では、<schemaversion>要素は以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	1 and only 1

データタイプ: <schemaversion>要素は文字列として表現される。XML Data Type: xs:string.

SCORM では<schemaversion>要素は以下の制約付きの語彙トークンを含む：

- **CAM 1.3:** この制約付きの語彙トークンは、コンテンツパッケージが SCORM コンテンツアグリゲーションモデルバージョン 1.3 に準拠して作られたことを示す

例：

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
  </metadata>
</manifest>
```

コード例3-4

3.4.1.5. {Meta-data}

メタデータは適切なメタデータスキーマを利用してマニフェストに挿入される[3]。 SCORM コンテンツモデル要素を記述するのにメタデータを利用するなら、ADL は、最低限 IEEE LOM メタデータスキーマを利用する事を強く推奨する。必要であれば、他の様々なメタデータスキーマを利用することがある（例えば Dublin Core）。このメタデータは、パッケージ全体（コンテンツアグリゲーションメタデータ）を表す。マニフェストにメタデータを挿入するには、いくつかのメカニズムがある。メタデータは、XML への拡張 (inline meta-data)によって、マニフェストに挿入されることが可能である。ADL は、スタンドアロン XML ファイルを参照するために、namespace された要素（「3.4.1.5.2< location>要素」参照）を提供する。<manifest>の子の {Meta-data} はオプションである（説明されたメカニズムの一つを利用して 0 回以上現れる可能性がある）。下記の例は、LOM 要素のインライン XML 拡張の利用を示す。

3.4.1.5.1. XML 拡張を利用するメタデータ

定義により、XML は、他の定義された名前空間から要素および属性を導入して拡張できる。メタデータは、このタイプの XML 拡張メカニズムを利用することによってコンテンツパッケージマニフェストに挿入される。拡張要素を XML ドキュメントに追加するにはいくつかの方法がある。

1. 特定の接頭辞を用いて XML 要素の名前空間を定義する。この方法では、特定の名前空間は、要素中で xmlns:<prefix>シンタックスを利用して、定義されなければならない。一般的に、これらの接頭辞は、XML 要素のルートノードで定義されるが、これは要件ではない。接頭辞と名前空間は、拡張要素が名前空間の宣言の前で用いられない限り、どの要素でも定義することができる（コード例 Illustration 3-5 参照）。

例：

```
<manifest xmlns:lom = "http://ltsc.ieee.org/xsd/LOM">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
    <lom:lom>
      <lom:general>
        <lom:title>
          <lom:string language="en-US">Title for the Package</lom:string>
        </lom:title>
      </lom:general>
      <lom:metaMetadata>
        <lom:metadataSchema>LOMv1.0</lom:metadataSchema>
        <lom:metadataSchema>SCORM_CAM_v1.3</lom:metadataSchema>
      </lom:metaMetadata>
    </lom:lom>
  </metadata>
</manifest>
```

コード例3-5

2. 特定の接頭辞なしで XML 要素の名前空間を定義する。この方法では、名前空間は、利用される箇所定義される。この方法を利用することにより、この要素およびその各子要素は特定の名前空間からのものであることがシンタックスによって示される（コード例3-6参照）。

例：

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
    <lom xmlns="http://ltsc.ieee.org/xsd/LOM">
      <general>
        <title>
          <string language="en-US">Title for the Package</string>
        </title>
      </general>
      <metaMetadata>
        <metadataSchema>LOMv1.0</metadataSchema>
        <metadataSchema>SCORM_CAM_v1.3</metadataSchema>
      </metaMetadata>
    </lom>
  </metadata>
</manifest>
```

コード例3-6

3.4.1.5.2. <location>要素

<location>要素は、SCORM コンテンツモデル要素を記述しているメタデータが見つかる可能性のある場所を指定する手段を提供する。これは URI であることがある。これは、IMS コンテンツパッケージング仕様に対する ADL の namespace 要素拡張である。メタデータ作成者はコンテンツパッケージでメタデータを表現する二つのオプションを持つ。作成者は、メタデータレコードのロケーションを表現するのに<location>要素を利用するか、もしくは前述の通りマニフェストファ

イル内にメタデータインラインをおくかのどちらかを利用できる。この値は、xml:base 値の利用によって影響される。xml:base の利用要件とガイダンスに関する詳細情報は、「3.4.4.1: XML Base 属性の扱い」参照。

XML 名前空間: http://www.adlnet.org/xsd/adlcp_v1p3

XML 名前空間接頭辞: adlcp

XML バインディング表現: <location>

SCORM 要件: SCORM では、<adlcp:location>要素は以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or More
リソース	0 or More

データタイプ: <adlcp:location>要素は文字列として表される。文字列の SPM は 2000 文字である。XML Data Type: xs:string.

属性:

- なし

要素:

- なし

例

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
    <adlcp:location>course/metadata/course.xml</adlcp:location>
  </metadata>
</manifest>
```

コード例3-7

3.4.1.6. <organization> 要素

<organizations>要素はコンテンツパッケージにおける一つ以上の構造すなわちオーガニゼーションを記述する [3].

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <organizations>

SCORM 要件: SCORM では, <organizations>要素に関して, 全てのマニフェストは以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	1 and only 1

SCORM では, リソースパッケージを構築するとき, この要素はマニフェスト内で空の要素 (<organizations/>)として表現されなければならない. コンテンツアグリゲーションパッケージを構築するとき, この要素は, 最低一つの <organization>サブ要素を含まなくてはならない.

データタイプ: <organizations>要素は親要素である. 親要素は自身に関連付けられた値を持たない. 親要素は他の要素/属性のコンテナとしての役割を果たす. <organizations>要素は以下の要素/属性を含む:

属性:

- default (必須 - コンテンツアグリゲーションパッケージに対して) - default 属性は利用するデフォルトオーガニゼーションを識別する. この要素の値は, <organizations>要素の直接の子孫である<organization>要素の識別子属性を参照しなければならない.
XML Data Type: xs:IDREF.

要素:

- <organization>

例:

```
<organizations default="TOC1">
  <organization identifier="TOC1">
    <title>Introduction to SCORM for LMS Vendors</title>
    <!--organizations structure placed here -->
  </organization>
  <organization identifier="TOC2">
    <title>Introduction to SCORM for Content Vendors</title>
    <!--organizations structure placed here -->
  </organization>
</organizations>
```

コード例3-8

3.4.1.7. <organization> 要素

<organization>要素は、ある特定の階層オーガニゼーションを記述する[3]。コンテンツオーガニゼーションは、<organization>要素によって定義される。コンテンツオーガニゼーションは、概念的な用語である。コンテンツオーガニゼーションは、レッスン、モジュール、チャプター等である。コンテンツオーガニゼーションが何を定義するかは、オーガニゼーションのカリキュラムの分類による。<organization>要素はIMS SSではアクティビティを表す。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <organization>

SCORM 要件: SCORM では、全てのマニフェストは、<organizations>要素に関して以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 or More
リソース	0

リソースパッケージに対して、この要素は現れない。<organizations>要素（親）は、空であることを要求される。

データタイプ: <organization>要素は親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素/属性のコンテナとしての役割を果たす。<organization>要素は、以下の要素/属性を含む：

属性:

- **identifier**（必須）- オーガニゼーションに対するマニフェスト内で一意の識別子[3]。この値は一般的に編集者もしくは編集ツールが提供する。XML Data Type: xs:ID.
- **structure**（オプション）- オーガニゼーションの形を記述する[3]。structure 属性のデフォルト値は、指定されない場合、hierarchical である。値の SPM は 200 文字である。XML Data Type: xs:string.
- **adlseq:objectivesGlobalToSystem**（オプション、デフォルト = true）- この属性は、シーケンシング情報（「5.1.1 <sequencing> 要素」参照）で定義されたマップされたグローバル共有学習目標は、学習者とコンテンツオーガニゼーションにグローバルか（False）、LMS 内で学習者が存在する限り全てのコンテンツオーガニゼーションに対してグローバルか（True）、のどちらかであることを示す。XML Data Type: xs:boolean.

要素:

- <title>
- <item>
- <metadata>
- <imsss:sequencing>

例：

```
<organizations>
  <organization identifier="TOC1">
    <title> Introduction to SCORM for LMS Vendors </title>
    <item identifier="ITEM1" identifierref="RESOURCE1" isvisible="true">
      <title>SCORM Run-Time Environment Requirements</title>
    </item>
    <item identifier="ITEM2" identifierref="RESOURCE2" isvisible="true">
      <title>LMS Conformance Requirements</title>
    </item>
  </organization>
</organizations>
```

コード例3-9

3.4.1.8. <title> 要素

<title>要素はオーガニゼーションのタイトルを記述する[3]。この要素は学習者がどのオーガニゼーションを選ぶか決定するのを助けるために利用できる。オーガニゼーションが何を表すかにより、このタイトルはコース、モジュール、レッスンなどに対して利用される。

XML 名前空間: http://www.imsglobal.org/xsd/imsdp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <title>

SCORM 要件: SCORM では、全てのマニフェストは、<title>要素に関して以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	0

リソースパッケージではこの要素は現れない。<organizations>要素は空でなくてはならない。

データタイプ: <title>要素は文字列要素で表現される。文字列の SPM は 200 文字である。XML Data Type: xs:string.

例：

```
<organization identifier="TOC1">
  <title>Introduction to the SCORM</title>
</organization>
```

コード例3-10

3.4.1.9. <item>要素

<item>要素はオーガニゼーションの階層構造を表わすノードである[3]。<item>要素はコンテンツオーガニゼーション中のアクティビティを表す。<item>要素はオーガニゼーションの構造内でノードを記述する。<item>要素は他の<item>要素の中でネストし、何層にも繰り返されることが可能である。<item>要素のこの構造は、コンテンツオーガニゼーションを形成し学習コンテンツの部分間の関係性を記述する。

<item>要素は、他の<item>要素のコンテナとしての役割を果たすことも、リーフノードとしての役割を果たす事もある。<item>がリーフノードであれば、<item>は<resource>要素を参照する。<item>要素が親要素であれば、<item>自身が<resource>要素を参照することは許されない（リーフ<item>要素だけがリソースを参照することを許されている）。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <item>

SCORM 要件: SCORM では、全てのマニフェストは、<item>要素に関して以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージングアプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 or More
リソース	0

リソースパッケージではこの要素は現れない。<organizations>要素は空でなくてはならない。

データタイプ: <item>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<item>要素は以下の要素/属性を含む：

属性:

- **identifier** (必須) - identifier 属性は、アイテムに対するマニフェスト内で一意の識別子である[3]。XML Data Type: xs:ID.
- **identifierref** (オプション) - identifierref 属性は、リソースセクションもしくは (サブ) マニフェスト中の identifier への参照である[3]。

ADL Note: IMS Global Consortium, Inc.,はIMS コンテンツパッケージ仕様の新しいバージョンの開発を行っている。IMS が解決しようとしている大きな課題のひとつは、(サブ) マニフェストに関して、その使用法、使用上の要求、XML シンタックスの要求などである。現時点で、IMS の作業が終わるまで、ADL は (サブ) マニフェストを使用しないよう推奨する。(サブ) マニフェストに関する質問、懸案事項、新たな推奨項目は ADL に送られたい。

Identifierref は、(同一マニフェスト内の) <resource>の identifier、(マニフェスト内の (サブ) マニフェスト内の) <resource>の identifier、もしくは (サブ) マニフェストの identifier を参照することができる。(サブ) マニフェストは、ファイルの最終的なロケーションを決定するのに利用される。Identifierref が無い場合、オーガ

ニゼーション内のこのエントリーに関連付けられたコンテンツが無いと仮定される。値の SPM は 2000 文字である。XML Data Type: xs:string.

- isvisible (オプション) - isvisible 属性は、パッケージの構造が表示あるいは提示されたときに、このアイテムが表示されるかどうかを示す。存在しなければ、値はデフォルトで True になる[3]。値は、自身が定義されたアイテムのみに影響し、そのアイテムの子もしくはアイテムに関連付けられたリソースには影響しない。XML Data Type: xs:boolean.
- parameters (オプション) - parameters 属性は、起動時にリソースに渡される静的パラメータを有する。Parameters 属性は、<resource>要素を参照する<item>要素に対してだけ利用されるべきである。値 SPM は 1000 文字である。XML Data Type: xs:string.

parameters 属性値の許容される書式は

- #<parameter>
- <name>=<value>(&<name>=<value>)*(<parameter>)
- ?<name>=<value>(&<name>=<value>)*(<parameter>)

でなくてはならず、ここで、

- <parameter>, <name> および <value> は実装定義の文字列値である
- = は<name> と <value> の対を分離するために要求される
- & は<name> と <value> の複数の対を分離するために要求される
- (&<name>=<value>)* は 0 ないしそれ以上の<name> と <value> の対が連結できることを示す。

parameters 属性値で使用される文字は URL エンコードされる必要がある。RFC 3986 で URL エンコードの要求条件が定義されている。

要素:

- <title>
- <item>
- <metadata>
- <adlcp:timeLimitAction>
- <adlcp:dataFromLMS>
- <adlcp:completionThreshold>
- <imsss:sequencing>
- <adlnav:presentation>

例:

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true"
    parameters="?width=500&#038;length=300">
    <title>Content 1</title>
  </item>
</organization>
```

コード例 3-11

3.4.1.10. <title>要素

<title>要素はアイテムのタイトルを記述する[3]。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: `imscp`

XML バインディング表現: `<title>`

SCORM 要件: SCORM では、全てのマニフェストは、`<title>`要素に関して以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	0

リソースパッケージではこの要素は現れない。 `<organizations>`要素は空でなくてはならない。従って、`<item>`もしくは`<title>`は提供されない。

データタイプ: `<title>`要素は文字列要素で表現される。文字列の SPM は 200 文字である。XML Data Type: `xs:string`.

例:

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
    <title>Content 1</title>
  </item>
</organization>
```

コード例3-12

3.4.1.11. `<item>`要素

`<item>`要素は、任意の階層にぶら下がることできる。これは一般的にアグリゲーションのコンテンツ構造に基づく。 `<item>`要素は、`<item>`要素の子として 0 回以上現れることが可能である。 `<item>`要素の詳細はセクション 3.4.1.9 参照。

3.4.1.12. `<metadata>` 要素

`<metadata>`要素はアイテムを記述するメタデータを含む[3]。 SCORM は`<item>`を記述するのに利用されるメタデータを、アクティビティメタデータと定義する。「4.5.1.3: アクティビティメタデータ」参照。 `<metadata>`要素は、アクティビティを記述するメタデータのルートノードとみなされる。これは、アクティビティに関する全てのメタデータは`<metadata>`要素の子として定義されることを意味する。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: `imscp`

XML バインディング表現: `<metadata>`

SCORM 要件: SCORM では, 全てのマニフェストは, `<metadata>`要素に関して以下の繰り返し要件を遵守するという要件をつけている:

SCORM コンテンツパッケージングアプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0or1
リソース	0

リソースパッケージに対してこの要素は現れない. `<organizations>`要素は空でなくてはならない.

データタイプ: `<metadata>`要素は, 親要素である. 親要素は, 自身に関連付けられた値を持たない. 親要素は, 他の要素/属性のコンテナとしての役割を果たす. `<metadata>`要素は, 以下の要素/属性を含む:

属性:

- なし

要素:

- `{Meta-data}` - 詳細は「3.4.1.5: `{Meta-data}`」参照

ADL Note: これは, 「3.4.1.2 `<metadata>`要素」で定義された`<metadata>`要素とは違うものである. IMS コンテンツパッケージング仕様は, `<manifest>`要素の子として定義された`<metadata>`要素には`<schema>`および`<schemaversion>` 要素だけを許している.

例:

```
<organization>
  <item>
    <title>The organization title</title>
    <metadata>
      <adlcp:location>lesson1/lesson1MD.xml</adlcp:location>
    </metadata>
  </item>
</organization>
```

コード例3-13

3.4.1.13. `<timeLimitAction>`要素

`<timeLimitAction>`要素は, アクティビティの現在の試行に許された最長時間が超過したときに取られるアクションを記述する. 全ての時間追跡と時間制限アクションは SCO によってコントロールされる.

この要素は, ADL で定義された IMS コンテンツパッケージング仕様への拡張である. この要素は SCO を参照するリーフ`<item>`要素の子としてのみ必要に応じて現れる. SCO リソースを参照する`<item>`要素だけが, `<timeLimitAction>` 要素を含むことができる.

LMS は `cmi.time_limit_action` データモデル要素(SCORM ランタイム環境ブック参照[2])を初期化するのに、提供されていれば`<timeLimitAction>`要素の値を利用する。コンテンツ開発者が時間制限アクションを定義すると、SCO は時間切れに基づく全ての動作に責任があることになる。

XML 名前空間: `http://www.adlnet.org/xsd/adlcp_v1p3`

XML 名前空間接頭辞: `adlcp`

XML バインディング表現: `<timeLimitAction>`

SCORM 要件: SCORM では、全てのマニフェストは、`<timeLimitAction>`要素に対して以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージングアプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or 1
リソース	0

リソースパッケージに対して、この要素は現れない。`<organizations>`要素は空でなくてはならない。

データタイプ: `<timeLimitAction>`要素は文字列として表現される。文字列は以下の制約付きの文字列トークンのどれか一つであることを要求される：

- **exit,message:** 学習者は SCO を強制的に終了させられる。SCO は学習者に学習試行に許された最長時間が超過したことを示すメッセージを提供しなくてはならない
- **exit,no message:** 学習者はメッセージなしで SCO を強制終了させられる
- **continue,message:** 学習者は SCO を継続することを許される。SCO は学習者に学習試行に許された最長時間が超過したことを示すメッセージを提供しなくてはならない
- **continue,no message:** 学習者は学習試行に許された最長時間を超過したが、学習者にメッセージは与えられず、強制終了もされない

この特性が SCO 内で利用されると、SCO は時間切れに影響する時間を追跡し、(適切な場合)時間切れを示すメッセージを提供することになる。

例：

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
    <title>Content 1</title>
    <adlcp:timeLimitAction>exit,no message</adlcp:timeLimitAction>
  </item>
</organization>
```

コード例 3-14

3.4.1.14. `<dataFromLMS>` 要素

`<dataFromLMS>`要素は、`<item>`が表わすリソース (SCO) が起動後に利用する初期化データを提供する。このデータは、LMS には関係なく、SCO に対してだけ機能的な意味を持つ。この要素は、起動中に SCO が必要とするパラメータ(query string parameters)として利用してはならない。このタイプの機能が必要ならば、開発者は、SCO リソースを参照するアイテムの `parameters` 属性を利用しなくてはならない。

この要素は、IMS コンテンツパッケージング仕様への ADL で定義された拡張である。この要素は SCO を参照するリーフ<item>要素の子としてのみ必要に応じて現れる。SCO リソースを参照するこれらの<item>要素だけが<dataFromLMS>要素を含むことができる。

LMS は cmi.launch_data データモデル要素(SCORM ランタイム環境ブック参照[2])を初期化するために、提供されれば<dataFromLMS>要素の値を利用する。

XML 名前空間: http://www.adlnet.org/xsd/adlcp_v1p3

XML 名前空間接頭辞: adlcp

XML バインディング表現: <dataFromLMS>

SCORM 要件: SCORM では、全てのマニフェストは、<dataFromLMS>要素に関して以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージングアプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or 1
リソース	0

リソースパッケージに対してこの要素は現れない。<organizations>要素は空でなくてはならない。

データタイプ: <dataFromLMS>要素は文字列要素として表現される。文字列の SPM は 4096 文字である。

例:

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
    <title>Content 1</title>
    <adlcp:dataFromLMS>Some SCO Information</adlcp:dataFromLMS>
  </item>
</organization>
```

コード例 3-15

3.4.1.15. <completionThreshold>要素

<completionThreshold>要素は、<completionThreshold>が定義された<item>が参照する SCO リソースが利用することが可能な閾値を定義する。この要素は、IMS コンテンツパッケージング仕様への ADL で定義された拡張である。この要素はリーフ<item>要素の子としてのみ必要に応じて現れる。SCO リソースを参照するこれらの<item>要素だけが、<completionThreshold>要素を含むことができる。

LMS は cmi.completion_threshold データモデル要素(SCORM ランタイム環境ブック参照[2])を初期化するのに、提供されれば<completionThreshold>要素の値を利用する。

XML 名前空間: http://www.adlnet.org/xsd/adlcp_v1p3

XML 名前空間接頭辞: adlcp

XML バインディング表現: <completionThreshold>

SCORM Requirements: SCORM places a requirement that all manifests shall adhere to the following multiplicity requirements for the <completionThreshold> element:

SCORM 要件: SCORM では、全てのマニフェストは、<completionThreshold>要素に関して以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or 1
リソース	0

リソースパッケージに対してこの要素は表現されない。<organizations>要素は空でなくてはならない。

データタイプ: <completionThreshold>要素は、0.0 から 1.0 の間の小数値として表現される。

例：

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
    <title>Content 1</title>
    <adlcp:completionThreshold>0.75</adlcp:completionThreshold>
  </item>
</organization>
```

コード例3-16

3.4.1.16. <sequencing>要素

「5.1.1 <sequencing>要素」参照。

3.4.1.17. <presentation>要素

「5.2.1.1 <presentation>要素」参照。

3.4.1.18. <metadata> 要素

<metadata>要素は、オーガニゼーションを記述するメタデータである[3]。SCORM は、<organization>を記述するのに利用されるメタデータを、コンテンツオーガニゼーションメタデータと定義する。SCORM コンテンツオーガニゼーションメタデータアプリケーションプロファイルの要件については、「4.5.1.2: コンテンツオーガニゼーションメタデータ」参照。
<metadata> 要素は、コンテンツオーガニゼーションを記述するメタデータのためのルートノードとみなされる。これは、コンテンツオーガニゼーションのための全てのメタデータは、<metadata>要素の子として定義されることを意味する。

XML 名前空間: http://www.imsglobal.org/xsd/imsdp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <metadata>

SCORM 要件: SCORM では、全てのマニフェストは、<metadata>要素に関して以下の繰り返し要件を満たさなくてはならない：

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or 1
リソース	0

リソースパッケージに対して、この要素は出現しない。<organizations>要素は空でなくてはならない。

データタイプ: <metadata>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<metadata>要素は以下の要素/属性を含む：

属性:

- なし

要素:

- {Meta -data} - 詳細は「3.4.1.5: {Meta -data}」参照

ADL Note: これは、「3.4.1.2 <metadata>要素」で定義された<metadata>要素とは違うものである。IMS コンテンツパッケージング仕様は、<manifest>要素の子として定義された<metadata>要素には<schema>および<schemaversion> 要素だけを許している。

例:

```
<organization>
  <title>Introduction to SCORM</title>
  <item identifier="ITEM1" identifierref="RESOURCE1" isvisible="true">
    <title>SCORM Run-Time Environment Requirements</title>
  </item>
  <metadata>
    <adlcp:location>activities/activity1MD.xml</adlcp:location>
  </metadata>
</organization>
```

コード例 3-17

3.4.1.19. <sequencing> 要素

「5.1.1: <sequencing> 要素」参照。

3.4.1.20. <resources> 要素

<resources>要素はリソースへのリファレンスの集合である。<resources>要素に含まれる個々の<resource>要素の順序もしくは階層についての仮定はない[3]。

XML 名前空間: http://www.imsglobal.org/xsd/imsdp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <resources>

SCORM 要件: SCORM では , 全てのマニフェストは, <resources>要素に対して以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	1 and only 1
リソース	1 and only 1

データタイプ: <resources>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<resources>要素は以下の要素/属性を含む:

属性:

- xml:base (オプション) - xml:base 属性はマニフェストに含まれるコンテンツファイルに対する相対パスを提供する[3]。この要素の利用は W3C の XML Base [7] ワーキングドラフトで定義される。値の SPM は 2000 文字である。XML Data Type: xs:anyURI.

要素:

- <resource>

例:

```
<manifest>
  <metadata/>
  <organizations/>
  <resources>
    <resource identifier="RESOURCE1" type="webcontent" href="lesson1.htm">
      <file href="lesson1.htm"/>
    </resource>
    <resource identifier="RESOURCE2" type="webcontent" href="intro1.htm">
      <file href="intro1.htm"/>
    </resource>
    <resource identifier="RESOURCE3" type="webcontent" href="content1.htm">
      <file href="content1.htm"/>
    </resource>
    <resource identifier="RESOURCE4" type="webcontent" href="summary1.htm">
      <file href="summary1.htm"/>
    </resource>
  </resources>
</manifest>
```

コード例3-18

3.4.1.21. <resource>要素

<resource>要素は, リソースへの参照である[3]。SCORM 内で定義された 2 つの主要なリソースのタイプがある:

- SCO
- アセット

XML 名前空間: http://www.imsglobal.org/xsd/ims_cp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <resource>

SCORM 要件: SCORM では、全てのマニフェストは、<resource>要素に関して以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or More
リソース	0 or More

リーフ<item>要素はリソース (SCO リソースまたはアセットリソース) を参照しなくてはならない。<item>がリソースを参照すると、このリソースは学習者への配信および起動のために指定される対象となる。<item>が<resource>を参照するならば、リソースは以下の要件をみたさなければならない:

- Type 属性が webcontent に設定される
- adlcp:scormType が sco もしくは asset に設定される
- href 属性が要求される

データタイプ: <resource>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<resource>要素は以下の要素/属性を含む:

属性:

- identifier (必須) - identifier 属性は、リソースの識別子を表し、このリソースを含むマニフェストファイルの範囲内で一意である[3]。この識別子は、一般的には、マニフェスト開発時に編集者もしくは編集ツールによって提供される。XML Data Type: xs:ID.
- type (必須) - type 属性はリソースのタイプを示す[3]。値の SPM は 1000 文字である。XML Data Type: xs:string.
- href (オプション) - href 属性は、URL (Uniform Resource Locator) の参照である[3]。Href 属性は、このリソースのエントリーポイントもしくは起動ポイントを表す。外部 fully qualified URL も許される。この値は、xml:base 値に影響される。xml:base 利用要件およびガイダンスについては、「3.4.4.1 XML Base 属性の扱い」参照。値の SPM は 2000 文字である。XML Data Type: xs:string.
- xml:base (オプション) - xml:base 属性はマニフェストに含まれるファイルの相対パスを提供する。この要素の利用法は W3C の XML Base ワーキングドラフトで定義される。値の SPM は 2000 文字である。XML Data Type: xs:anyURI.
- adlcp:scormType (必須) - adlcp:scormType 属性は SCORM リソースのタイプを定義する。これは、IMS コンテンツパッケージング情報モデルへの ADL の拡張である。XML Data Type: xs:string。文字列は制限され、以下の文字列トークンのうちどちらかである (sco もしくは asset)。sco はリソースが SCO リソースであることを示し、asset はリソースはアセットリソースであることを示す。
- adlcp:persistState (オプション) - adlcp:persistState 属性は、学習者試行から学習者試行ヘデータを保持する手段を提供する。SCORM ランタイム環境ブック[2]で記述されたように、新規の学習者試行が開始されると、新たなランタイムデータが提供される。adlcp:persistState 属性が定義され true に設定されると、古い学習試行のデータが新しい学習者試行を初期化するのに利用される。XML Data Type: xs:boolean。デフォルト値は、属性が提供されない場合、false である。

Elements:

- <metadata>
- <file>
- <dependency>

例:

```
<resources>

  <resource identifier="R_A2" type="webcontent" adlcp:scormType="sco"
            adlcp:persistState="true" href="sco1.html">
    <file href="sco1.html"/>
  </resource>

<!-- adlcp:persistState is not defined, default value of false should be applied
-->
  <resource identifier="R_A3" type="webcontent" adlcp:scormType="sco"
            href="sco3.html">
    <file href="sco3.html"/>
  </resource>

  <resource identifier="R_A5" type="webcontent" adlcp:scormType="asset"
            href="pics\distress_sigs_add.jpg">
    <file href="pics\distress_sigs_add.jpg"/>
  </resource>
</resources>
```

コード例 3-19

3.4.1.22. <metadata>要素

<metadata>要素は、リソースを記述するメタデータである [3]。SCORM は、<resource>を記述するのに利用されるメタデータを SCO メタデータもしくはアセットメタデータと定義する。これは、リソースの SCORM タイプ(adlcp:scormType)による。「4.5.1 SCORM 要素へのメタデータの対応付け」参照。<metadata> 要素は、リソースを記述するメタデータに対してルートノードとみなされる。これは、リソースに対して全てのメタデータは<metadata>要素の子として定義されることを意味する。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <metadata>

SCORM 要件: SCORM では、全てのマニフェストは、<metadata>要素に関して以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or 1
リソース	0 or 1

データタイプ: <metadata>要素は親要素である。親要素は、自身に関連付けられた値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<metadata>要素は、以下の要素/属性を含む:

属性:

- なし

要素:

- {Meta-data} - 詳細は「3.4.1.5 {Meta-data}」参照

ADL Note: これは、「3.4.1.2 <metadata>要素」で定義された<metadata>要素とは違うものである。IMS コンテンツパッケージ仕様は、<manifest>要素の子として定義された<metadata>要素には<schema>および<schemaversion> 要素だけを許している。

例:

```
<resources>
  <resource identifier="R_A2" type="webcontent" adlcp:scormType="sco"
    href="sco1.html">
    <file href="sco1.html"/>
    <metadata>
      <adlcp:location>resources/resource1MD.xml</adlcp:location>
    </metadata>
  </resource>
</resources>
```

コード例3-20

3.4.1.23. <file>要素

<file>要素は、このリソースが依存するファイルを列挙したものである[3]。この要素は、あるリソースに対する各ファイルに対して必要に応じてリピータされる。この要素は、リソースを構築するのに利用されるファイルのセットの詳細を記述するインベントリースystemとしての役割を果たす。<file>要素はコンテンツパッケージにローカルなファイルを表す。コンテンツパッケージにローカルな（コンテンツパッケージに物理的に置かれている）全てのファイルに対して、<file>要素は、そのファイル自身が利用されるリソースに関連のあるファイルを表すのに利用される。リソースがパッケージにローカルと指定されると、リソース自身が<file>要素として指定される。<resource>の起動ロケーション（<resource>の href 値）は、ファイルの href として利用される。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表示: <file>

XML バインディング表現: <file>

SCORM 要件: SCORM では、全てのマニフェストは、<file>要素に対して以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or More
リソース	0 or More

データタイプ: <file>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<file>要素は以下の要素/属性を含む:

属性:

- href (必須) -href 属性は、ファイルのロケーションを指定する[3]。この値は xml:base 値の利用によって影響される。xml:base 利用要件およびガイダンスについては、「3.4.4.1 XML Base 属性の扱い」参照。値の SPM は 2000 文字である。XML Data Type: xs:string。

要素:

- <metadata>

例:

```
<resource identifier="R_A2" type="webcontent" adlcp:scormType="sco"
  href="sco1.html">
  <file href="assets/image1.gif"/>
  <file href="sco1.html"/>
  <file href="assets/common/APIWrapper.js"/>
</resource>
```

コード例 3-21

3.4.1.24. <metadata>要素

<metadata>要素は、ファイルを記述するメタデータである [3]。SCORM は、<file>を記述するのに利用されるメタデータを、アセットメタデータと定義する。「4.5.1.5 アセットメタデータ」参照。<metadata> 要素は、アセットを記述するメタデータに対してルートノードである。これは、アセットに対して全てのメタデータは、<metadata>要素の子として定義されることを意味する。

XML 名前空間: http://www.imsglobal.org/xsd/imsdp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <metadata>

SCORM 要件: SCORM では、全てのマニフェストは、<metadata>要素に関して以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or 1
リソース	0 or 1

データタイプ: <metadata>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素/属性のコンテナとしての役割を果たす。<metadata>要素は以下の要素/属性を含む:

属性:

- なし

要素:

- {Meta-data} - 詳細は「3.4.1.5 {Meta-data}」参照

ADL Note: これは、「3.4.1.2 <metadata>要素」で定義された<metadata>要素とは違うものである。IMS コンテンツパッケージング仕様は、<manifest>要素の子として定義された<metadata>要素には<schema>および<schemaversion> 要素だけを許している。

例:

```
<resource identifier="R_A2" type="webcontent" adlcp:scormType="sco"
  href="sco1.html">
  <file href="assets/image1.gif">
    <metadata>
      <adlcp:location>assets/asset1.xml</adlcp:location>
    </metadata>
  </file>
  <file href="sco1.html"/>
  <file href="assets/common/APIWrapper.js"/>
</resource>
```

コード例3-22

3.4.1.25. <dependency>要素

<dependency>要素は、このリソースが依存するファイルのリソースを指定する[3]。
<dependency>が参照するリソースは、<dependency>を含むリソースが依存する複数ファイルに対するコンテナとしての役割を果たすことができる。

XML 名前空間: http://www.imsglobal.org/xsd/imscp_v1p1

XML 名前空間接頭辞: imscp

XML バインディング表現: <dependency>

SCORM 要件: SCORM では、全てのマニフェストは、<dependency>要素に対して以下の繰り返し要件を満たさなくてはならない:

SCORM コンテンツパッケージング アプリケーションプロファイル	マニフェスト繰り返し要件
コンテンツアグリゲーション	0 or More
リソース	0 or More

データタイプ: この要素は空の要素として表現される。<dependency>要素は属性だけを含む。

属性:

- identifierref (必須) - identifierref 属性は、<resource>(同一パッケージ内)または(サブ)マニフェストの identifier 属性を参照し、従属リソースの最終的なロケーションを特定するのに利用される。値の SPM は 2000 文字である。XML Data Type: xs:string.

要素:

- なし

例：

```
<resources>
  <resource identifier="R_A2" type="webcontent" adlcp:scormType="sco"
    href="sco1.html">
    <file href="sco1.html"/>
    <dependency identifierref="R_A5"/>
  </resource>
  <resource identifier="R_A5" type="webcontent" adlcp:scormType="asset"
    href="pics\distress_sigs_add.jpg">
    <file href="pics\distress_sigs_add.jpg"/>
  </resource>
</resources>
```

コード例3-23

3.4.1.26. <manifest> 要素

セクション「3.4.1.1 <manifest> 要素」参照.

ADL Note: IMS Global Consortium, Inc.,はIMS コンテンツパッケージ仕様の新しいバージョンの開発を行っている。IMS が解決しようとしている大きな課題のひとつは、(サブ) マニフェストに関して、その使用法、使用上の要求、XML シンタックスの要求などである。現時点で、IMS の作業が終わるまで、ADL は(サブ) マニフェストを使用しないよう推奨する。(サブ) マニフェストに関する質問、懸案事項、新たな推奨項目はADL に送られたい。

3.4.1.27. <sequencingCollection>要素

「5.1.12 <sequencingCollection>要素」参照.

3.4.2. コンテンツパッケージマニフェストの拡張

IMS コンテンツパッケージ仕様は、コミュニティがマニフェストを通して独自の名前空間が付けられた要素をおくことを認めている。SCORM はいくつかの拡張によりこれを継承する。これら拡張は、様々な要件を満たし、SCORM で記述された各種の仕様および規格のアプリケーションプロファイルを支援しなくてはならない。これら拡張は、3つの別々のXML スキーマ定義(XSD) ファイルによって定義される。これらのファイルは：

- adlcp_v1p3.xsd: SCORM コンテンツパッケージング拡張を記述する。これらの拡張要素および属性は、http://www.adlnet.org/xsd/adlcp_v1p3 名前空間内で定義される。ADL は、その名前空間の要素を示すために 名前空間と adlcp: 接頭辞を予約している。これらの拡張要素の詳細は「3.4.1 マニフェストファイル」で説明されている
- adlseq_v1p3.xsd: SCORM シーケンシング拡張を記述する。これらの拡張要素および属性は、http://www.adlnet.org/xsd/adlseq_v1p3 名前空間内で定義される。ADL は、その名前空間の要素を示すために 名前空間と adlseq: 接頭辞を予約している。これらの拡張要素の詳細は「3.4.1 マニフェストファイル」で説明されている
- adlnav_v1p3.xsd: SCORM ナビゲーション拡張を記述する。これらの拡張要素および属性は、http://www.adlnet.org/xsd/adlnav_v1p3 名前空間内で定義される。ADL は、その

名前空間の要素を示すために 名前空間と adlnav: 接頭辞を予約している。これらの拡張要素の詳細は「3.4.1 マニフェストファイル」で説明されている

マニフェストを組織で定義された拡張で拡大すると、セマンティック相互運用可能性が低くなる事がある。拡張を認識しない組織は、どのように拡張に対応するか分からず、拡張を無視することがある。マニフェストを拡張する必要性が生じ、セマンティック相互運用可能性を高いレベルで保つには、ADL は、相互運用可能な拡張の構築に、コミュニティが共通見解を持つように協力することを推奨する。

3.4.3. コンテンツパッケージマニフェストの Href の扱い

“href” は、コンテンツパッケージマニフェストで指定された<file>もしくは<resource>のロケーションを記述するのに利用される。このロケーションは内部 URL もしくは外部 URL のどちらかである。IMS コンテンツパッケージングバージョン 1.1.3 最終仕様によれば、href の値は RFC 2396 [8] で表現されるルールに従って作成される。

3.4.3.1. XML Base 属性の扱い

XML Base [7] は、コンテンツパッケージのファイルにリンクする相対 URL の（絶対 URL への）変換において、ドキュメントのベース URL を明示的に使用するのに用いるデータ構造である。URL は XML Base 属性の接頭辞を付けることができる。これにより、編集者ないし編集ツールがベースディレクトリを指定したりオフセットを設けることが可能となり、そのベースディレクトリの URL を毎回使うたびに繰り返す必要性が無くなる。

デフォルトベースディレクトリは、パッケージのロケーションである。この概念は、「パッケージに相対」（“relative to the package”）として知られる。このデフォルトベースディレクトリを明示的に上書きする唯一の方法は、パッケージに対して外部にある絶対パスをもつファイルを参照することである。ファイルが、パッケージに対して外部になれば、XML Base 値は単にデフォルトベースディレクトリにオフセットを加える。XML Base 属性は以下のどちらかである：

- 相対 URL（パッケージのルートからのオフセットを記述する。Course/Lesson/など）、もしくは、
- 外部 URL（パッケージの外部。http://www.adlnet.org/content/など）

XML Base 値の最後にはスラッシュが要求される。コンテンツパッケージでローカルファイルを参照するとき、XML Base を含む URL の先頭にスラッシュ（“/”）で始まるべきではない。RFC 2396 で定義されたように、先頭のスラッシュで始まるパスは、そのファイルの絶対的なパスを示す。先頭のスラッシュはローカルホストのルートを示す。これを考慮して、先頭のスラッシュの使用は、誤まった解釈を最小化しポータビリティを向上するために許されていない。

IMS コンテンツパッケージング XML バインディング仕様は、<manifest>、<resources>および<resource>要素での XML Base 属性の利用を認めている。

XML Base 属性が<manifest>要素に存在すれば、<manifest>要素の子要素内で見つかる全ての URL は、実際の href 値を作成するために XML Base 値を用いなくてはならない。これは<resource>および<file>要素に対する href 値および<adlcp:location>要素に保持された値を含む。

```

<manifest xml:base="Course/">
  <organizations>
    <organization>
      <item identifier="ID1" identifierref="R_ID1"></item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="R_ID1"
      href="Lesson01/Topics/index.htm"></resource>
  </resources>
</manifest>

```

コード例 3-24

<manifest>要素で XML Base 属性が用いられ、href が<manifest>要素の子階層内に存在するため、コード例 3-24 で示されるリソースに対する実際の href は **Course/Lesson01/Topics/index.htm** となる。

<resources>要素に XML Base 属性が存在すれば、<resources>要素の子要素内の全ての URL は実際の href 値を作るために XML Base 値を用いなくてはならない。

```

<manifest>
  <organizations>
    <organization>
      <item identifier="ID1" identifierref="R_ID1"></item>
    </organization>
  </organizations>
  <resources xml:base="Course/Lesson01/">
    <resource identifier="R_ID1" href="Topics/index.htm"></resource>
  </resources>
</manifest>

```

コード例 3-25:

<resources>要素で XML Base 属性が用いられ、href が<resources>要素の子階層内に存在するため、コード例 3-25 で示されるリソースに対する実際の href は **Course/Lesson01/Topics/index.htm** となる。

XML Base 属性が<resource>要素に存在すれば、<resource>要素の子要素内の全ての URL は実際の href 値を作るために XML Base 値を用いなくてはならない。

```

<manifest>
  <organizations>
    <organization>
      <item identifier="ID1" identifierref="R_ID1">
      </item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="R_ID1" xml:base="Course/Lesson01/Topics/"
      href="index.htm">
    </resource>
  </resources>
</manifest>

```

コード例 3-26

<resource>要素で XML Base 属性が用いられ、href が<resource>要素の子階層内に存在するため、コード例 3-26 で示される通りソースに対する実際の href は **Course/Lesson01/Topics/index.htm** となる。

XML Base 属性の組み合わせが、マニフェスト全体に利用されると、XML Base 属性の値は、実際の URL を形成するために、これらの要素の階層の順序に追加される。<manifest>要素の XML Base 値が最初に、次に<resources>要素の XML Base 値、次に<resource>要素の XML Base 値、次に href 属性値が来る。

```
<manifest xml:base="Course/">
  <organizations>
    <organization>
      <item identifier="ID1" identifierref="R_ID1"></item>
    </organization>
  </organizations>
  <resources xml:base="Lesson01/">
    <resource identifier="R_ID1"
      href="index.htm" xml:base="Topics/">
    </resource>
  </resources>
</manifest>
```

コード例 3-27

<manifest>、<resources>、および、<resource>要素で XML Base 属性が用いられ、href が <resource>要素の属性として存在するため、コード例 3-27 で示されるリソースに対する実際の href は **Course/Lesson01/Topics/index.htm** となる。

3.4.3.2. URL エンコーディングとデコーディング

いくつかの状況で、ファイルもしくはリソースのロケーションを定義するのに利用される URL は、符号化される必要がある。RFC 2396 は、URL を符号化するルールと要件を定義する。RFC 2396 のセクション 2.2 から 2.4 で、いつどのように URL を符号化し復号するかが記述されている。いくつかの文字は、URL でデリミタとしての目的を持ち、その目的に使われているときには取り除くことができない。これらの文字は以下の通りである。

- "/" URL のスキーマ部分もしくは URL のパス部分でセパレータとして
- ":" URL のスキーマ部分で
- "#" アンカー値に対する先頭のキャラクターとして
- "&" パラメータのセパレータとして
- "?" URL のパス部分とパラメータのセパレータとして
- "=" パラメータ値とパラメータ名のセパレータとして
- "%" エスケープインディケータとして

これらの文字の一つが、上記と異なる目的で URL 中に存在する場合、それらの文字はエンコードされなければならない（エスケープされなければならない）。これを考慮して、"Course/Lesson/Module/Resources/bar.html" といった完全な URL を取り出し、ECMAScript のエスケープ機能を利用してエスケープするのは誤りである。なぜなら、それによって、URL の解釈に要求されるパスセパレータのスラッシュがエスケープされた文字に変更されてしまうからである。一方、URL の一部として含まれるパラメータの値がスラッシュ文字を含む場合、その文字はエスケープされなければならない。従って、エスケープは URL の構成要素を集める前に、エスケープされるべきセグメントをエスケープして、次にそれらをエスケープすべきでないデリミタで組み立てることによって行われなければならない。

さらに、URL がマニフェストに含まれているとき、2 重符号化は行なうべきでない。こうして、リソースに対する href 属性の値は、ブラウザがリソースを起動するのに必要なフォーマットの有効な URL を含む文字列となる。

さらに、<item>に対してパラメータが指定されたら、これらは URL の中で用いるために、適切にエスケープされなくてはならない。例えば、以下のパラメータが SCO に必要と仮定すると：

- "?ratio=3/4&scale=100&label=Gilbert & Sullivan"

上記の例は、不正な文字を含むので有効なパラメータ値ではない。“3/4”の“/”は、URL の一部でなく、パスのセパレータとして利用されないので、エスケープされる必要がある。また“Gilbert & Sullivan”の“&”も、パラメータ間のセパレータとして利用されないのでエスケープされる必要がある。これらのパラメータを正しくエスケープした形は：

- "ratio=3%2F4&scale=100&label=Gilbert %26 Sullivan"

下記の形は、キャラクターが2重エスケープされており、正しくエスケープされていない

- "ratio=3%252F4&scale=100&label=Gilbert %2526 Sullivan"

マニフェストは XML で実装されるので、XML のエスケープに関するルールにも従われなければならない。例えば、XML1.0 規格によれば、“&”はこの形で現れるのはコメント、処理指示、ないし、CDATA セクションの中である。上記の例のように、それ以外で必要であれば、数値文字参照もしくは文字列 (“%26” or “&”) を利用してエスケープされなければならない。以下の形がアイテム要素のパラメータ属性で利用する正しい値である：

- "ratio=3%2F4%26scale=100%26label=Gilbert %26 Sullivan"

“=”もしくは“ ”といった他のキャラクターをエスケープするのも認められている：

- "ratio%3D3%2F4%26scale%3D100%26label%3DGilbert%20%26%20Sullivan"

3.4.3.3. Parameters 属性の扱い

コンテンツオブジェクトが正しく作動するために、コンテンツオブジェクトが起動時に情報を要求する場合がある。この情報は、しばしば起動パラメータ（もしくはクエリ文字列）と呼ばれる。マニフェストでクエリ文字列を表すメカニズムが、現在二つある。

- **オプション 1:** <resource>もしくは<file> href 属性の 1 部として。コンテンツ開発者は href の 1 部としてクエリ文字列をおくことができる。この例は：

```
<resource href="foo.html?Topic=1">
  <!-- resource information -->
</resource>
```

コード例 3-28

- **オプション 2:** <item>の parameters 属性を利用して。コンテンツ開発者には、<resource>を参照する<item>の parameters 属性に、クエリ文字列すなわち起動パラメータをおくという選択肢がある。<resource>要素の href 属性は、リソースを起動するのに利用される URL である。この URL は、パッケージ中のファイルを指すこともそうでないこともある。<file>要素の href はファイル名を指定し、必要であれば、パッケージがインストールされたルートディレクトリに対する相対インストールパスを指定する。これらは、必ずしも同一でないで、冗長ではない。例えば、<resource>要素の href は"scos/foo.html#xyz"といったものになる一方、<file>の href は"scos/foo.html"ということがある。

マニフェストは、リソースに対する起動ファイルも含め、パッケージに含まれる全てのファイルのインベントリであることにも注意しなくてはならない。言い換えると、<file>要素はインベントリエントリである。<resource>要素は特定のファイルの集まりをどのように利用するか（すなわちどのように外部リソースにアクセスするか）を指定し、<organization>の<item>要素はコンテンツオーガニゼーションの一つ以上の場所で、どのようにリソースを利用するかを指定する。

<item>が<resource>を参照する場合、当該の<resource>は、リソースを起動するための href エントリを含まなくてはならない。Parameters 属性は、静的パラメータとして定義され、起動時にリソースに渡される。これにより、異なったアイテムから異なった目的で同じ<resource>を参照することが可能となる。

例：

```
<manifest>
  <!-- For brevity, elements not in question or part of the -->
  <!-- example were kept simple and incomplete -->
  <organizations>
    <organization>
      <item identifier="I01" identifierref="R_I01"
        parameters="?Topic=1">
      </item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="R_I01" href="foo.htm">
    </resource>
  </resources>
</manifest>
```

コード例3-29

オプション1では、href 属性でパラメータが定義され、各リソースはマニフェストで繰り返されなければならない

マニフェスト内で起動パラメータを表す方法が文法的に多々あるので、IMS コンテンツパッケージング仕様では、リソース要素の href 属性とパラメータを作成するアルゴリズムを詳細に記述している。

```
While first char of parameters is in "&"
  Clear first char of parameters
If first char of parameters is "#"
  If URL contains "#" or "?"
    Discard parameters
  Done processing URL
If URL contains "?"
  Append "&" to the URL
Else
  Append "?" to the URL
Append parameters to URL
```

コード例3-30

3.5. SCORM コンテンツパッケージアプリケーションプロファイル

SCORM コンテンツパッケージアプリケーションプロファイルは、SCORM のコンテキスト内で、IMS コンテンツパッケージング仕様がどのように適用されるかを記述する。アプリケーションプロファイルは、実装者に対して実践的なガイダンスを提供し、また、他の規格や仕様を統合し、相互運用可能性をより確実にするために SCORM によって追加された要件を定義する。IMS コンテンツパッケージング仕様は SCORM コンテンツパッケージの基礎として用いられる。しかし SCORM は、IMS 仕様に定義されたものに加えて、各パッケージに十分な情報を含めるための追加要件を要求する。これにより、SCORM 対応システムが、他の SCORM 対応システムで 사용되는パッケージをインポートおよびエクスポートすることが可能となる。

SCORM では、オブジェクトベースの学習コンテンツの一般化された枠組みを定義するコンテンツアグリゲーションモデル（「セクション 2.1 SCORM コンテンツモデル要素」）が導入される。部要素はアセット、SCO およびコンテンツオーガニゼーションである。現在二つの SCORM コンテンツパッケージアプリケーションプロファイルがあり、これはどのようにコンテンツアグリゲーションモデル要素をパッケージするかを記述する：

- リソースパッケージ
- コンテンツアグリゲーションパッケージ

以下のセクションではアプリケーションプロファイル、SCORM による制約、推奨されるベストプラクティスを説明する。

3.5.1. リソースパッケージ

SCORM リソースパッケージアプリケーションプロファイルは、オーガニゼーション、学習コンテキスト、もしくは、カリキュラム分類を提供することなく、アセットと SCO をパッケージ化するメカニズムを定義する。学習リソースをパッケージ化することで、交換のための共通媒体が提供される。リソースパッケージアプリケーションプロファイルは、システムからシステムへ SCO とアセットを移すために用いなくてはならない。リソースパッケージでは定義されたオーガニゼーションがないため、論理的なコンテンツ構造が定義されない。構造が定義されないため、このタイプのパッケージは LMS から学習者に配信することができない。SCORM リソースパッケージは、単に学習システム間で交換することができる学習リソースの集合である。

多くのケースで、アセットリソースもしくは SCO リソースは、単一ファイルから成り立つが、アセットや SCO が、複数ファイルから成り立つケースがある。SCORM リソースパッケージアプリケーションプロファイルは、アセットや SCO は単一でも複数のファイルでもパッケージすることを認めている。また、アセットや SCO は、ローカルでパッケージに含まれることもあれば、外部を参照することもある。ローカルにパッケージされたファイルは、全体のパッケージの中に物理的なファイルとして含まれる。外部参照する場合は、アセットや SCO は、パッケージ内の物理的なファイルに含まれず、代わりに URL によって参照される。

以下の図は、いくつかのリソースパッケージを示す。例は、サンプル `imsmanifest.xml` インスタンスおよびアセットと SC0 がどのように表現されるかを示す。図 3.5.1a は、`imsmanifest.xml` インスタンスで `<file>` 要素として提示されたアセットの例を示す。

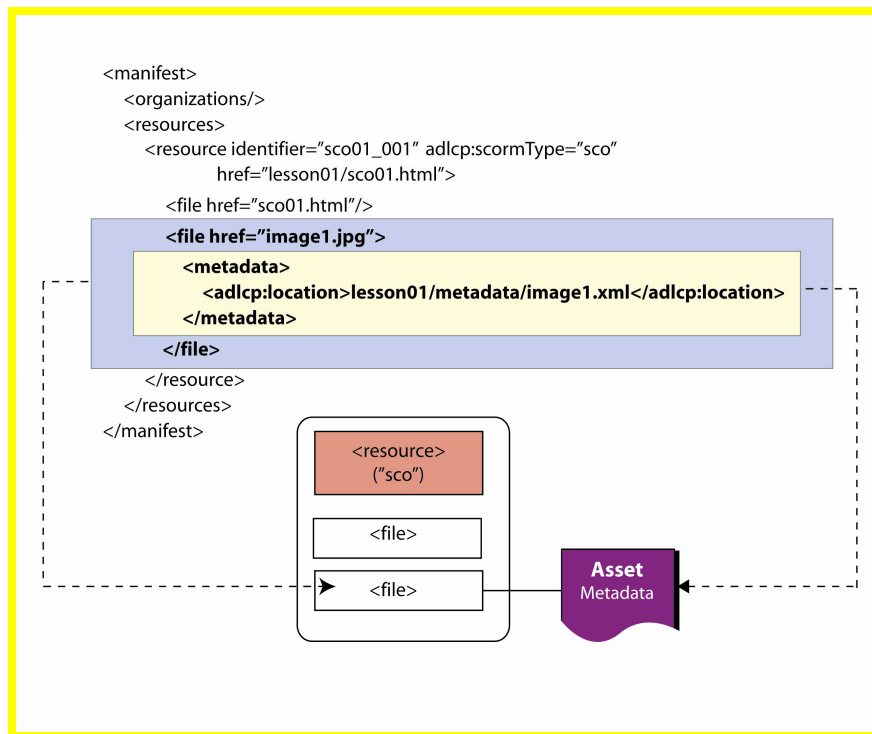


図3.5.1a: `<file>` 要素として提示されたアセットの例

図 3.5.1b は、`imsmanifest.xml` インスタンスで `<resource>` 要素として表現されたアセットの例を示す。

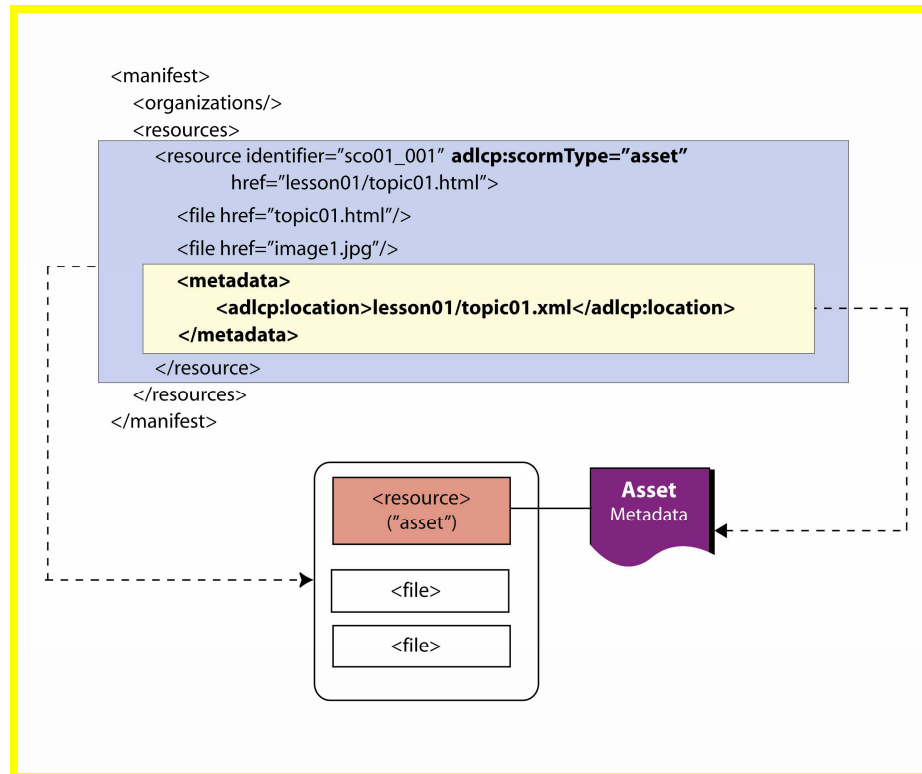


図 3.5.1b : <resource> 要素として提示されたアセットの例

図 3.5.1c は, imsmanifest.xml インスタンスで<resource>要素として表現された SCO の例を示す.

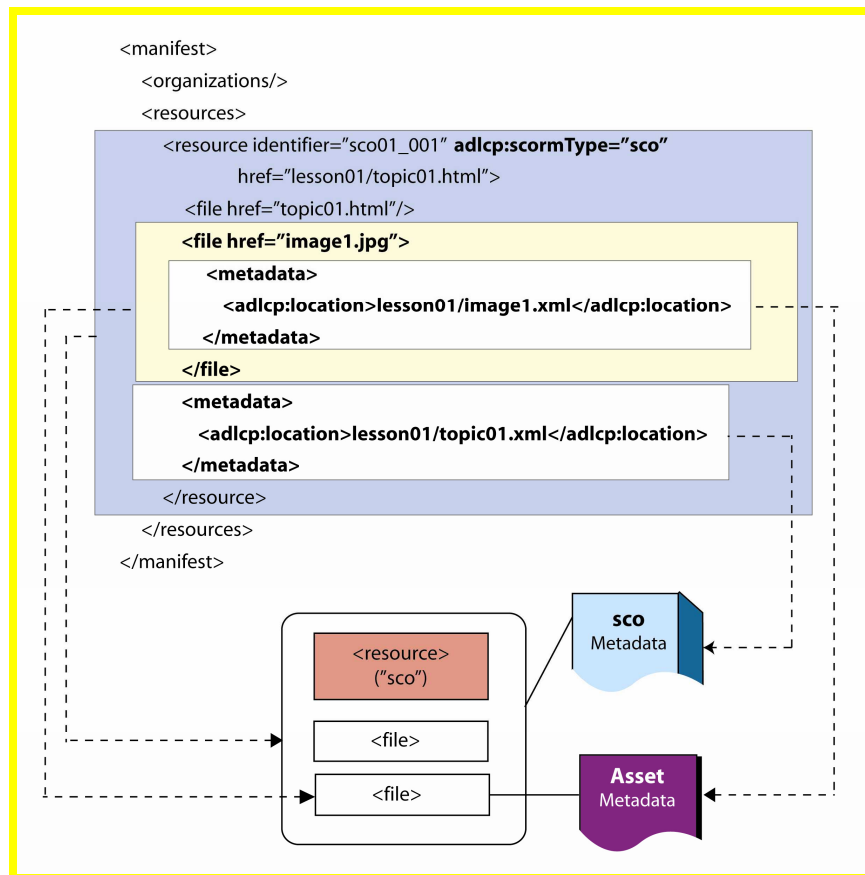


図3.5.1c : <resource>要素として提示されたSCO の例

3.5.2. コンテンツアグリゲーションパッケージ

SCORM はコンテンツオーガニゼーションの構造に、いかなる要件も強いことはない。個々のコンテンツ開発者は、各自にとって価値のあるいかなる構造にでもコンテンツを統合できる。IMS コンテンツパッケージ仕様[3]は、ADL に必要な情報の殆どと、その他の情報を捉えるために ADL 拡張が追加される論理的な場所を含む枠組みを提供する。さらに、IMS パッケージモデルは、学習リソースを配信するのに必要な全ての物理ファイルを整理してバンドルし、コンテンツパッケージ内の物理ファイルとして含まれない外部参照されたリソースも含めて、一つ以上の学習リソースに属するファイル間の関係を指定する明確な方法を提供する。コンテンツアグリゲーションアプリケーションプロファイルは学習リソースとコンテンツ構成をバンドルするのに用いなくてはならない。これは、完全なコース、モジュール、レッスンなどをバンドルするのに用いられるアプリケーションプロファイルである。コンテンツアグリゲーションパッケージの主な目的は、エンドユーザにコンテンツを配信するために用いることである。

IMS コンテンツパッケージ仕様は、複数の学習リソースを組織化する構成と、個々の学習リソースを分離することも可能にし、同じ学習リソースを異なった文脈で一回以上利用することを認めている。SCORM はファイルをパッケージ化し、構造を提供するメカニズムを定義している。

図 3.5.2a は、imsmanifest.xml インスタンスで表現されたコンテンツオーガニゼーションの例を示す。

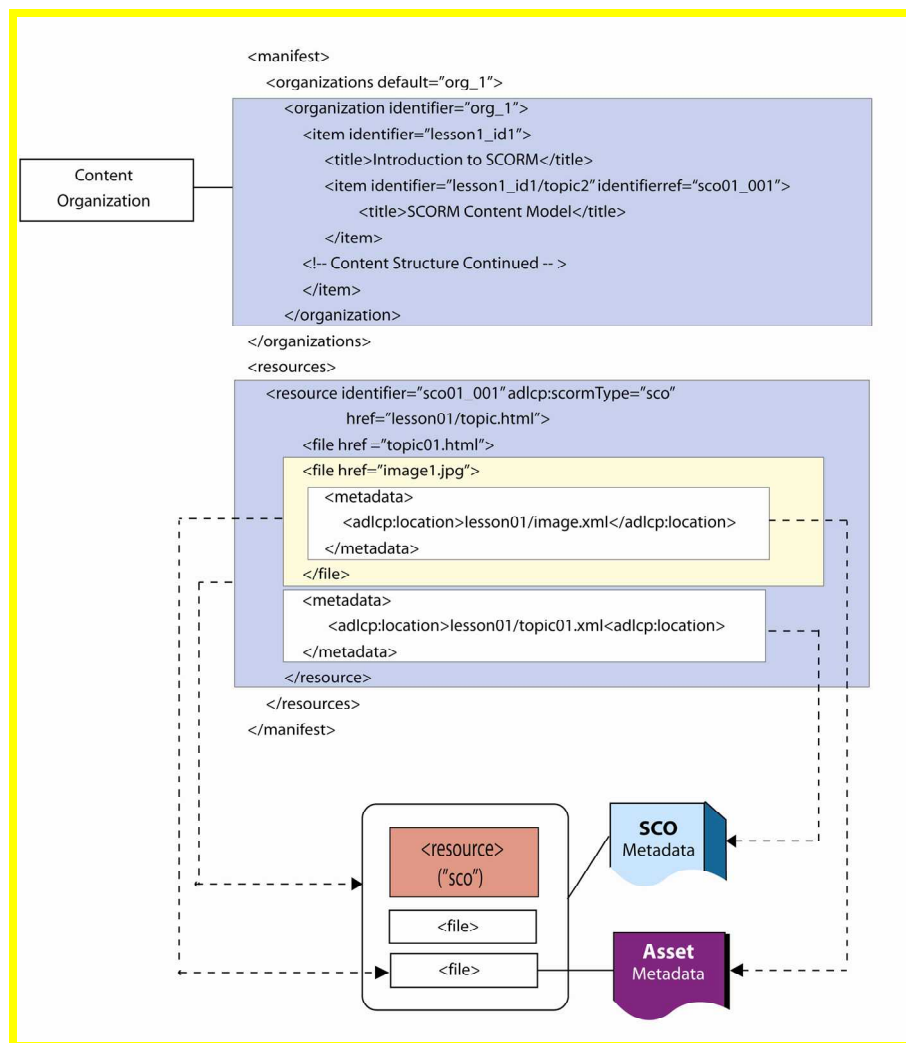


図3.5.2a : コンテンツアグリゲーションパッケージ

3.5.3. SCORM コンテンツパッケージアプリケーションプロファイル要件

表 3.5.3a は各々のコンテンツパッケージアプリケーションプロファイルの要件を定義する。各プロファイルに対して、各コンテンツパッケージングマニフェスト要素/属性に対応する要件が列挙されている。

- “M” は要素/属性が必須であることを示す
- “O” は要素/属性がオプションであることを示す
- “NP” は要素/属性が認められないことを示す

要素は XML Binding を用いて示されている (XML 表記<element_name>を用いて)。属性は標記なしで示される (例えば 1.1 identifier は<manifest>要素の属性)。通番システムは、IMS コンテンツパッケージング仕様に基づく。

表3.5.3a: SCORM コンテンツパッケージアプリケーションプロファイルマニフェスト要素要件

No.	要素	リソースパッケージ	コンテンツアグリゲーションパッケージ
1	<manifest>	M	M
1.1	Identifier	M	M
1.2	Version	O	O
1.3	xml:base	O	O
1.4	<metadata>	O	O
1.4.1	<schema>	O	O
1.4.2	<schemaversion>	O	O
1.4.3	{Meta-data}	O	O
1.5	<organizations>	M	M
1.5.1	Default	NP	M
1.5.2	<organization>	NP	M
1.5.2.1	Identifier	NP	M
1.5.2.2	Structure	NP	O
1.5.2.3	adlseq:objectivesGlobalToSystem	NP	O
1.5.2.4	<title>	NP	M
1.5.2.5	<item>	NP	M
1.5.2.5.1	Identifier	NP	M
1.5.2.5.2	Identifierref	NP	O
1.5.2.5.3	<title>	NP	M
1.5.2.5.4	Isvisible	NP	O
1.5.2.5.5	Parameters	NP	O
1.5.2.5.6	<item>	NP	O
1.5.2.5.7	<metadata>	NP	O
1.5.2.5.7.1	{Meta-data}	NP	O
1.5.2.5.8	<adlcp:timeLimitAction>	NP	O
1.5.2.5.9	<adlcp:dataFromLMS>	NP	O
1.5.2.5.10	<adlcp:completionThreshold>	NP	O
1.5.2.5.11	<imsss:sequencing>	NP	O
1.5.2.5.12	<adlnav:presentation>	NP	O
1.5.2.6	<metadata>	NP	O
1.5.2.6.1	{Meta-data}	NP	O
1.5.2.7	<imsss:sequencing>	NP	O

1.6	<resources>	M	M
1.6.1	xml:base	O	O
1.6.2	<resource>	O	O
1.6.2.1	identifier	M	M
1.6.2.2	type	M	M
1.6.2.3	href	O	O
1.6.2.4	adlcp:scormType	M	M
1.6.2.5	adlcp:persistState	O	O
1.6.2.6	xml:base	O	O
1.6.2.7	<metadata>	O	O
1.6.2.7.1	{Meta-data}	O	O
1.6.2.8	<file>	O	O
1.6.2.8.1	href	M	M
1.6.2.8.2	<metadata>	O	O
1.6.2.8.2.1	{Meta-data}	O	O
1.6.2.9	<dependency>	O	O
1.6.2.9.1	identifierref	M	M
1.7	<manifest>	O	O
1.8	<imsss:sequencingCollection>	NP	O

3.6. ベストプラクティスおよび実践的なガイドライン

以下のセクションは、コンテンツパッケージの開発に関して推奨されるベストプラクティスおよび実践的なガイドラインが説明されている。これらのベストプラクティスは適合要件とはみなされない。

3.6.1. 単一コースに対する複数オーガニゼーション

コンテンツパッケージは、そのリソースに対する複数のオーガニゼーションの表現を許している。同じリソースが、異なった学習者に対して適用される異なったコンテンツオーガニゼーションで利用されることがある。例えば、初心者に対しては学習ユニットを飛ばす機能を持たせないでコンテンツを順々に進捗させることが望ましいこともあれば、進んだ学習者には自分が経験したい学習ユニットだけを選択して記憶を新たにするためにコンテンツを利用させたいこともある。複数オーガニゼーションは異なる理由のために異なった方法で、一組のリソースを構成するために用いることができる。複数オーガニゼーション要素の利用はユースケースには理想的である。しかし、コンテンツ開発者が、複数の固有のコースをパッケージ化して移動したければ、これらのコースは別々の（サブ）マニフェストで作成されるべきである。

3.6.2. 再利用のための学習コンテンツのパッケージング

マニフェストの範囲は流動的である。マニフェストはコースの文脈外で自身で存在しうるコンテンツオーガニゼーションの一部（教材オブジェクト）、コンテンツオーガニゼーション全体、コンテンツオーガニゼーションの集合、などを記述することができる。この決断は、コンテンツ開発者に任されており、自分の望むようにコンテンツを集約および分解するように記述することができる。

一般的なルールは、コンテンツパッケージが常に単一のトップレベルマニフェストを含み、このマニフェストが必要に応じて一つ以上の（サブ）マニフェストを持つことである。トップレベルマニフェストは常にパッケージを記述する。ネストした（サブ）マニフェストは、コース、学習オブジェクトなど自身が対象とするレベルのコンテンツを記述する。

例えば、一つのコンテンツオーガニゼーションからなるコンテンツの全てが密接につながり、コンテンツオーガニゼーションのコンテキスト外に出せる部分がない場合、コンテンツ開発者はそのコンテンツオーガニゼーションのリソースおよびオーガニゼーションを記述するのに単一のマニフェストを使うであろう。しかし、学習オブジェクトを作成し、これを他の学習オブジェクトと再結合させて異なったコース表現を作りたいコンテンツ開発者は、各学習オブジェクトをそれぞれ専用のマニフェストに記述し、それらのマニフェストをコンテンツオーガニゼーションを含む上位レベルのマニフェストに統合したいと思うであろう。

3.6.3. <dependency> 要素の利用

コンテンツパッケージで定義されたいくつかの学習リソースは、同一のセットのファイルを含むことがある。ファイルは、マニフェストで<file>要素として提示される。<dependency>要素は、

これらのファイルをグループ化するのに利用することができる。このシナリオで<dependency>要素を用いると、各リソースでの各ファイルセットに対する<file>要素の重複を少なくすることができる。このシナリオで<resource>要素は、ファイルのセットを集めるのに利用することができる。一旦<resource>がセットアップされると、ファイルに依存する他の全てのリソースは、<dependency>要素を利用してそのリソースを参照することが可能である。

```
<manifest>
  <organizations>
    <organization>
      <item identifier="ID1" identifierref="R_ID1"></item>
      <item identifier="ID2" identifierref="R_ID2"></item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="R_ID1" adlcp:scormType="sco" href="index_1.htm">
      <file href="index_1.htm"/>
      <file href="image1.jpg"/>
      <file href="image2.jpg"/>
      <file href="image3.jpg"/>
      <file href="apiWrapper.js"/>
    </resource>
    <resource identifier="R_ID2" adlcp:scormType="sco" href="index_1.htm">
      <file href="index.htm"/>
      <file href="image1.jpg"/>
      <file href="image2.jpg"/>
      <file href="image3.jpg"/>
      <file href="image4.gif"/>
      <file href="apiWrapper.js"/>
    </resource>
  </resources>
</manifest>
```

コード例3-31：ファイル共有

コード例 3-31：ファイル共有で、定義された二つのリソースが共通のファイルを共有している：

- image1.jpg
- image2.jpg
- image3.jpg
- apiWrapper.js

これらのファイルは、各リソースに対して<file>要素として繰り返される。上記のメソッドは、これらの<file>要素の繰り返しをなくすために利用することが可能である。

```

<manifest>
  <organizations>
    <organization>
      <item identifier="ID1" identifierref="R_ID1"></item>
      <item identifier="ID2" identifierref="R_ID2"></item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="R_ID1" adlcp:scormType="sco" href="index_1.htm">
      <file href="index_1.htm"/>
      <dependency identifierref="DEP_R_ID1"/>
    </resource>
    <resource identifier="R_ID2" adlcp:scormType="sco" href="index_1.htm">
      <file href="index.htm"/>
      <file href="image4.gif"/>
      <dependency identifierref="DEP_R_ID1"/>
    </resource>
    <resource identifier="DEP_R_ID1" adlcp:scormType="asset">
      <file href="image1.jpg"/>
      <file href="image2.jpg"/>
      <file href="image3.jpg"/>
      <file href="apiWrapper.js"/>
    </resource>
  </resources>
</manifest>

```

コード例 3-32

コード例 3-32 で、共通で利用されるファイルを保持するためにリソースが作られた。このリソースは固有の識別子を与えられた。新たに作成されたリソースはアセットである。href 属性はこのリソースには提供されない。このアセットは LMS によって起動されることはない（<item>の identifierref がリソースを参照することはない）。これらのファイルを共有するリソースは、identifierref 属性によって新たに作成されたリソースを参照する <dependency> 要素を含む。

セクション 4

SCORM メタデータ

このページは空白である。

4.1. SCORM メタデータ概要

現時点まで、SCORM は、コンテンツ開発に対する基本的な構築ブロック(SCORM コンテンツモデル部品)を記述してきた。また、SCORM は、これらの構築ブロックをどのようにコンテンツアグリゲーションに束ね、システムからシステムへ配信するのにどのようにパッケージ化するかも記述してきた。SCORM コンテンツモデル部品が、一旦、構築されると、これらの部品を一貫性のある形で記述するのが有用である。部品をメタデータで記述することは、システムを跨いで部品の検索検索および発見を可能にする。LMS は、コンテンツオーガニゼーション(コース、レッスン、モジュールなど)に関する学習者情報を与えるためにメタデータを利用することができる。また、メタデータは、ランタイム時に、学習者にどんなコンテンツモデル部品を配信するか決定するのに支援するのに利用されることも可能である。

このセクションは、SCORM コンテンツモデル部品へメタデータを適用する特定の要件およびガイドラインを提供する。このセクションで定義される SCORM メタデータアプリケーションプロファイルは、IEEE 1484.12.1 2002 Learning Object Meta-data (LOM) [11]規格および IEEE 1484.12.3 Draft Standard for Extensible Markup Language (XML) Binding for Learning Object Metadata Data Model [14]を直接参照する。IEEE は、およそ 64 のメタデータ要素を提供する。日々の利用にはこれで充分であろう。このセクションは、SCORM 文脈で、コンテンツアグリゲーションモデルで記述された部品を分類するために利用されるメタデータの中のどのデータ要素が必須かを定義する。SCORM は、IEEE 規格に十分に準拠するが、このセクションは、SCORM 部品を記述するのにメタデータを利用することに対して、追加の特定のガイダンスを提供する。SCORM は、SCORM コンテンツモデル部品の記述に対して IEEE LOM を利用することを強く推奨するが、他のメタデータスキーマメタデータスキーマも利用することは可能である。これらのメタデータスキーマは、システムに認識されることもされないこともある。

以下のセクションは、5 つの基本的なサブセクションに分類され、各々が異なった部分を SCORM メタデータに関して説明する：

- セクション 4.1: **メタデータ概要**。このセクションは、一般的な概要と LOM に関する基礎情報を提供する
- セクション 4.2: **メタデータ作成**。このセクションは、メタデータ作成に対する要件を定義する。このセクションは、IEEE に定義された要件の詳細、およびどのようにこれらの要件が、SCORM に影響するかを提供する。このセクションは、IEEE LOM 要件に準拠して XML インスタンスを作成する詳細を提供する
- セクション 4.3: **LOM XML スキーマバリデーションアプローチ**。このセクションは、IEEE Draft Standard for Extensible Markup Language (XML) Binding for Learning Object Metadata Data Model で開発された認証アプローチを説明する。認証アプローチは XML 認証要件に対して、ユーザーニーズによって異なったサポートを提供する。
- セクション 4.4: **メタデータ拡張**。このセクションは、IEEE Draft Standard for Extensible Markup Language (XML) Binding for Learning Object Metadata Data Model および SCORM によって定義された拡張機能を説明する。また、このセクションは、拡張を作成する賛否を議論する。
- セクション 4.5: **SCORM メタデータアプリケーションプロファイル**。このセクションは、SCORM 環境でメタデータをどのように実装するかに関して具体的なガイダンスを提供する。IEEE に定義された全ての要素は、オプションとみなされる。このセクションは、異

なった SCORM メタデータアプリケーションプロファイルの SCORM 必須要素およびこれらが SCORM 対応に対してどのように XML で表現されるかを定義する。

メタデータの目的は、共通名称を提供し学習リソースが共通の方法で記述されるようにすることである。メタデータは、カタログに収集されることも可能であり、自身が記述する学習リソースに直接組み込まれることも可能である。メタデータで記述された学習リソースは、システムチックに検索され、利用および再利用に検索される。

SCORM は、IEEE LOM メタデータ要素定義を、コンテンツアグリゲーションモデルで記述された SCORM コンテンツモデル部品に適用する。これらの部品は、SCORM CAM のメタデータの観点を定義する。

IEEE から SCORM CAM への標準化された定義のマッピングは、一般的な仕様と特定のコンテンツモデル間のミッシングリンクを提供する。以下のセクションは、SCORM CAM のメタデータに関する観点への IEEE 規格の適用を定義する。

4.2. LOM メタデータ作成

以下のセクションは、LOM XML メタデータ要素の概要を記述する。IEEE によれば、どの LOM メタデータ要素もオプションである。これは、XML メタデータインスタンスを構築するとき、開発者は、どの要素を利用するか選択できることを意味する。

ADL のキーとなるハイレベル要件のいくつかを満たすために、SCORM は、SCORM 対応メタデータ XML インスタンスで、どの要素が必須であるかに関する追加要件を課している。これらの追加要件は、オブジェクトをメタデータで記述する機能(要求される要素の一貫性のあるセットを利用して、一貫性のある方法で)を可能にし、レポジトリの学習オブジェクトを見つけ、他の文脈で利用する機能も可能にする。この要求される要素の一覧は、メタデータに表わされる SCORM コンテンツモデル部品(アセット、SCO、アクティビティ、コンテンツオーガニゼーション、コンテンツアグリゲーション)によって、異なる。これらの要素およびその利用要件の完全な一覧に関しては、セクション 4.5.2: *SCORM メタデータアプリケーションプロファイル要件* 参照。

1484.12.1 2002 情報モデル (今後 LOM 情報モデルと言及される)は、SCORM 対応メタデータを構築するのに利用可能なデータ要素を記述する。LOM 情報モデルで定義された要件と並んで、SCORM は、いくつかのタイプのメタデータインスタンスに対して、アプリケーションプロファイルを定義する。アプリケーションプロファイルの要件および定義は、セクション 4.5.2: *SCORM メタデータアプリケーションプロファイル要件* で見つけることができる。SCORM 対応メタデータは、セクション 4.4: *メタデータ拡張* で記述されるように、追加のデータ要素を含むことがある。

LOM 情報モデルは、9 カテゴリーに区分される。これらのカテゴリーは、LOM 情報モデルの定義に基づく。メタデータ要素の 9 カテゴリーは：

1. *General* カテゴリーは、SCORM コンテンツモデル部品全般の一般的な情報を記述するのに利用される
2. *Life Cycle* カテゴリーは、SCORM コンテンツモデル部品の過去と現在の状態に関連する特徴およびその進化過程で影響を与えたものを記述するのに利用される
3. *Meta-metadata* カテゴリーは、(SCORM コンテンツモデル部品を記述するレコードでなく)メタデータレコード自体の情報を記述する表わすのに利用される
4. *Technical* カテゴリーは、SCORM コンテンツモデル部品のテクニカルな要件や特徴を記述するのに利用される
5. *Educational* カテゴリーは、SCORM コンテンツモデル部品の教育的および学術的な特徴を記述するのに利用される
6. *Rights* カテゴリーは、SCORM コンテンツモデル部品の知的所有権や利用の条件を記述するのに利用される
7. *Relation* カテゴリーは、SCORM コンテンツモデル部品と他の対象となる部品の関係を定義する情報を記述する表わすのに利用される
8. *Annotation* カテゴリーは、SCORM コンテンツモデル部品の教育的利用へのコメントおよびそのコメントがいつ誰に作成されたかの情報を提供するのに利用される
9. *Classification* カテゴリーは、ある区分システムで SCORM コンテンツモデル部品がどこに区分されるかを記述するのに利用される

いくつかの要素は、繰り返しおよび/もしくはデータタイプを記述するのに、用語 SPM (smallest permitted maximum) を利用する。SPM は、メタデータを処理するアプリケーションが、最低どれだけの数のキャラクターを処理しなければならないかを示すが、その数を越えたものをサポートするのは自由である。

語彙タイプのデータタイプを持つ要素に対して、語彙が、「制限された」語彙か「ベストプラクティス」語彙であるかの追加の情報が提供される。「制限された」は、メタデータ要素は、一覧にされている語彙エントリーに制限されることを示す。「ベストプラクティス」は、SCORM がリストされた語彙エントリーをベストプラクティスとして利用することを推奨することを示す。

以下の表は、SCORM メタデータアプリケーションプロファイル(詳細はセクション 4.5 参照)要件を記述するのに利用される：

表3.4.1a: SCORM メタデータアプリケーションプロファイルテーブルフォーマット

SCORM メタデータアプリケーションプロファイル	メタデータ繰り返し要件
コンテンツアグリゲーション	<requirement> (SPM: x)
コンテンツオーガニゼーション	<requirement> (SPM: x)
アクティビティ	<requirement> (SPM: x)
SCO	<requirement> (SPM: x)
アセット	<requirement> (SPM: x)

左の欄は SCORM メタデータアプリケーションプロファイルである。この欄は、SCORM に定義されたアプリケーションプロファイルの種類を記述する。右の欄は *Meta-data* 繰り返し要件である。この欄はメタデータの XML 要素に対する繰り返し要件を定義する。

表3.4.1b: 繰り返し要件の説明

繰り返し要件	説明
1 and only 1	親要素内で 1 回だけ存在しなければならない
0 or More	親要素内で 0 回以上存在することができる
1 or More	親要素内で 1 回以上存在しなければならない
2 or More	親要素内で 2 回以上存在しなければならない
0 or 1	親要素内で 0 回か 1 回存在することができる

表 3.4.1b は、このセクションで利用される繰り返しのタイプを定義している。各タイプは簡単な説明を伴う。この表は、1 以上の繰り返しを持つ要素の SPM の定義もしている。SPM は、処理システムでサポートされなければならない要素の最小値を示す。

4.2.1. <lom> 要素

全てのメタデータインスタンスは、ルートノードとして<lom>を持つ。ルートノードは、SCORM コンテンツモデル部品を記述するのに利用されるメタデータを定義し始める。コンテンツパッケージにおかれるとき (セクション 4.5.1: *Associating Meta-data with SCORM Components* 参照), 全てのメタデータは、imsmanifest.xml ファイルの<imscp:metadata>要素(セクション 3.4.1: *Manifest File* 参照)内におかれる。<lom> root node は、上記に表わされた全てのカテゴリーを内部に閉じ込めている。9 カテゴリーの暗示された順序はない。子要素はどんな順序で現れても構わない。

全ての名前空間宣言は、<lom>要素内で宣言されるべきである。これは、メタデータへの拡張とみなされるどんな名前空間も含む。これは要件とはみなされないが、XML 仕様に基づき、ADL は、これをベストプラクティスとみなし、ベンダおよびツールに、この情報を提供するように勧めている。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <lom>

繰り返し要件: <lom>要素は、全ての SCORM コンテンツモデル部品を記述するのに SCORM が要求する重要な要素を含む。もし、LOM メタデータが SCORM コンテンツモデル部品を記述するのに用いられる場合、<lom>要素は1回だけ現れなくてはならない。 **ADL Note:** <lom>要素の繰り返し要件は、<lom>要素 (LOM インスタンスのルート要素) が、ある LOM インスタンスについて1回だけ現れなくてはならないことを示唆している。一つ以上の LOM インスタンスを、ある SCORM コンテンツモデル部品を記述するのに用いることができる。この場合、<lom>要素は、<imscp:metadata> element の直接の子孫として0回ないしそれ以上現れることができる (詳細については 3.4.1.2 <metadata> 要素参照)。

データタイプ: <lom>要素は、親要素である。親要素は、自身と関連付けられた値を持たない。親要素は、他の要素に対して、コンテナとしての役割を果たす。<lom>要素は、以下の子要素を含む:

- <general>
- <lifeCycle>
- <metaMetadata>
- <technical>
- <educational>
- <rights>
- <relation>
- <annotation>
- <classification>

例：

この例は、上記のコンセプトを表すのに利用される、9 カテゴリー要素は、簡素化のために、空の要素として提示されている。

```
<lom xmlns="http://ltsc.ieee.org/xsd/LOM">
  <general/>
  <classification/>
  <annotation/>
  <lifeCycle/>
  <technical/>
  <metaMetadata/>
  <educational/>
  <relation/>
  <rights/>
</lom>
```

コード例4-1

4.2.2. <general> 要素

General カテゴリーは、リソース全体を記述する一般的な情報を区分する。このケースでリソースとは、記述されたある特定の SCORM コンテンツモデル部品（アセット、SCO、アクティビティもしくはコンテンツオーガニゼーション）である。この一般的な情報は、特定の部品を記述するのに重要であるという観点から、キー情報として見られることがある。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <general>

繰り返し要件: <general>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <general>要素は、親要素である。親要素は、自身に関連づけられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<general>要素は、以下の子要素を含む：

- <identifier>
- <title>
- <language>
- <description>
- <keyword>
- <coverage>
- <structure>
- <aggregationLevel>

例：

```
<lom>
  <general>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/content/CO_01</entry>
    </identifier>
    <title>
      <string language="en">Title for the learning object</string>
    </title>
    <language>en</language>
    <description>
      <string language="en">Textual description</string>
    </description>
    <keyword>
      <string language="en">learning object</string>
    </keyword>
    <coverage>
      <string language="en">Circa, 16th century France</string>
    </coverage>
    <structure>
      <source>LOMv1.0</source>
      <value>atomic</value>
    </structure>
    <aggregationLevel>
      <source>LOMv1.0</source>
      <value>2</value>
    </aggregationLevel>
  </general>
</lom>
```

コード例4-2

4.2.2.1. <identifier> 要素

<identifier>要素は、SCORM コンテンツモデル部品を識別するグローバルに固有のラベルを割り当てるメカニズムを表す。多面的な学習コンテンツ開発に取り組むときに、グローバルに固有のラベルを割り当てるといふ観念は、重要である。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <identifier>

繰り返し要件: <identifier>要素は、0 ないし 1 回現れなくてはならない。<identifier>要素の SPM は 10 である。

データタイプ: <identifier>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<identifier>要素は以下の子要素を含む：

- <catalog> - optional
- <entry> - mandatory

例：

```

<lom>
  <general>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/content/C0_01</entry>
    </identifier>
  </general>
</lom>

```

コード例4-3

4.2.2.1.1. <catalog> 要素

<catalog>要素は、エントリに対する識別名称もしくはカタログ化スキームを掲示する。様々な利用可能なカタログ化システムがある。SCORMは、ある特定のカタログ化システムの利用を要求しない。組織は、自身の組織の慣行および方針を満たすカタログ化スキームを自由に選択するのである。いくつかのタイプのカタログ化システムは：

- Universal Resource Identifier (URI)
- Universal Resource Name (URN)
- Digital Object Identifier (DOI)
- International Standard Book Numbers (ISBN)
- International Standard Serial Numbers (ISSN)

<catalog>要素は、エントリを作成し管理するスキームを掲示する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <catalog>

繰り返し要件: <catalog>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <catalog>要素は、文字列要素として表現される。文字列は、1000 キャラクタのSPMを持つ（詳細はセクション4.2.11.1: 文字列データタイプ参照）。

例：

```

<lom>
  <general>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/content/C0_01</entry>
    </identifier>
  </general>
</lom>

```

コード例4-4

4.2.2.1.2. <entry>要素

<entry>要素は、学習オブジェクトを特定するカタログ化スキーム内の識別子の値を表す。

識別子は、様々なフォーマットを取り入れることが可能である。IEEE は、実際の識別子の値が文字列として表されることを要求する。組織は、自分の組織の慣行および方針を満たす固有の識別子を作成するどんなメカニズムも自由に選択できる。

以下の一覧は、識別子値（エントリ）の例である：

スキーム (<catalog>)	値 (<entry>)
Universal Resource Name	urn:ADL: 1345-GFGC-23ED-3321
Universal Resource Identifier	http://www.adlnet.org/content/C0_01
ADL Registry	2134-RF43-3233-FRI9-ACDA

ADL Note: ADL Registry は存在しない。これは、存在する可能性のある登録システムの一つのタイプの単なる例である。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <entry>

繰り返し要件: <entry>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <entry>要素の値は、文字列要素として表現される。文字列は、1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11.1: 文字列データタイプ参照）。

例:

```
<lom>
  <general>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/content/C0_01</entry>
    </identifier>
  </general>
</lom>
```

コード例 4-5

4.2.2.2. <title>要素

<title>要素は、学習オブジェクトに与えられた名称を表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <title>

繰り返し要件: <title>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <title>要素は、LangString 要素として表現される。LangString, 1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 2: LangString データタイプ参照）。

例：

```
<lom>
  <general>
    <title>
      <string language="en">Sharable Content Object Reference Model</string>
    </title>
  </general>
</lom>
```

コード例4-6

4.2.2.3. <language>要素

<language>要素は、第一人間言語もしくは SCORM コンテンツモデル部品内で意図されたユーザに
送信するのに利用される言語を表す。言語要素は、繰り返されるされることが可能である。これ
は、複数言語をサポートするように作られた部品を記述する機能を考慮している。

<language>要素に保持される値は、以下に従って表される：

Language = Langcode(“ -“ Subcode)*

Langcode - ISO 639:1988 で定義された通りに言語コードを表す。この値は必須である。

Subcode - ISO 3166-1997 で定義されたコードセットより国コードを表す。この値は、リピー
トされることがありオプションである。

例：

- “ en ”
- “ en -GB ”

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <language>

繰り返し要件：<language>要素は、0 ないしそれ以上現れなくてはならない。<language>要素の
SPM は 10 である。

データタイプ：<language>要素は、文字列要素として表現される。文字列は、1000 キャラクタ
ーの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。

例：

```
<lom>
  <general>
    <language>en</language>
    <language>fr</language>
  </general>
</lom>
```

コード例4-7

4.2.2.4. <description>要素

<description>要素は、メタデータで記述された SCORM コンテンツモデル部品のテキスト記述を表す。Description 要素は、部品の説明を考慮する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <description>

繰り返し要件: <description>要素は、0 ないしそれ以上現れなくてはならない。<description>要素の SPM は 10 である。

データタイプ: <description>要素は、LangString 要素として表現される。LangString は、2000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.2: LangString データタイプ参照)。

例:

```
<lom>
  <general>
    <description>
      <string language="en">Textual description of the learning
object</string>
    </description>
  </general>
</lom>
```

コード例4-8

4.2.2.5. <keyword>要素

<keyword>要素は、一般的なキーワードもしくは学習オブジェクトを記述するフレーズを定義するのに利用される。キーワードを作成するとき、作成者は、非常に簡明で SCORM 部品に固有の言葉やフレーズを選択すべきである。キーワード要素は、1 単語もしくはフレーズからなる。一つ以上のキーワードが望まれる場合、作成者は、キーワード要素の複数インスタンスを利用すべきである。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <keyword>

繰り返し要件: <keyword>要素は、0 ないしそれ以上現れなくてはならない。<keyword>要素の SPM は 10 である。

データタイプ: <keyword>要素は、LangString 要素として表現される。LangString は、1000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.2: LangString データタイプ参照)。

例：

```
<lom>
  <general>
    <keyword>
      <string language="en">learning object</string>
      <string language="nl">leerobject</string>
      <string language="fr">objet d'apprentissage</string>
    </keyword>
    <keyword>
      <string language="en">metadata</string>
      <string language="nl">metadata</string>
      <string language="fr">métadonnées</string>
    </keyword>
  </general>
</lom>
```

コード例4-9

4.2.2.6. <coverage> 要素

<coverage>要素は、SCORM コンテンツモデル部品が適用する時間、文化、地理もしくは地域を記述するのに利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <coverage>

繰り返し要件: <coverage>要素は、0 ないしそれ以上現れなくてはならない。<coverage>要素の SPM は 10 である。

データタイプ: <coverage>要素は、LangString 要素として表現される。LangString は、1000 キャクターの SPM を持つ（詳細はセクション 4.2.11.2: LangString データタイプ参照）。

例：

```
<lom>
  <general>
    <coverage>
      <string language="en">Circa, 16th century France</string>
    </coverage>
  </general>
</lom>
```

コード例4-10

4.2.2.7. <structure>要素

<structure>要素は、SCORM コンテンツモデル部品の基礎をなす組織構成を記述する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <structure>

繰り返し要件：<structure>要素は、0 ないし 1 回現れなくてはならない。

データタイプ：<structure>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン：SCORM は、<structure>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- **atomic:** 分割できないオブジェクト
- **collection:** 特定の関係のないオブジェクトのセット
- **networked:** 関係が特定されないオブジェクトのセット
- **hierarchical:** ツリー構造で関係が表されるオブジェクトのセット
- **linear:** 完全に整列されたオブジェクトのセット。例：「前」と「後」の関係で結ばれたオブジェクトのセット

例：

```
<lom>
  <general>
    <structure>
      <source>LOMv1.0</source>
      <value>atomic</value>
    </structure>
  </general>
</lom>
```

コード例 4-11

4.2.2.8. <aggregationLevel>要素

<aggregationLevel>要素は、学習オブジェクトの機能的な塊を記述する。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <aggregationLevel>

繰り返し要件：<aggregationLevel>要素は、0 ないし 1 回現れなくてはならない。

データタイプ：<aggregationLevel>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン：SCORM は、<aggregationLevel>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- **1:** 最小レベルのアグリゲーション、例：生メディアデータもしくはフラグメント
- **2:** レベル 1 学習オブジェクトの集合、例：レッスン
- **3:** レベル 1 学習オブジェクトの集合、例：コース
- **4:** 最大レベルの塊、例：認定に繋がる 1 セットのコース

例：

```
<lom>
  <general>
    <aggregationLevel>
      <source>LOMv1.0</source>
      <value>2</value>
    </aggregationLevel>
  </general>
</lom>
```

コード例 4-12

4.2.3. <lifeCycle>要素

Life Cycle カテゴリは、SCORM コンテンツモデル部品の過去と現在の状態およびその進化の過程で影響を与えた人に関する特徴を区分する。このカテゴリで収集される情報の典型的なタイプは、部品のステータス(部品は最終的な状態なのかドラフトフォーマットなのか等)、バージョン識別子、何らかの形で部品に影響を与えた個人もしくは団体の一覧などが含まれる。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <lifeCycle>

繰り返し要件: <lifeCycle>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <lifeCycle>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<lifeCycle>要素は、以下の子要素を含む：

- <version>
- <status>
- <contribute>

例:

```
<lom>
  <lifeCycle>
    <version>
      <string language="en">1.0 alpha</string>
    </version>
    <status>
      <source>LOMv1.0</source>
      <value>final</value>
    </status>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>author</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe FridayEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">A description for the date</string>
        </description>
      </date>
    </contribute>
  </lifeCycle>
</lom>
```

コード例 4-13

4.2.3.1. <version>要素

<version>要素は、SCORM コンテンツモデル部品のバージョンを記述する。部品は、ライフサイクル中に、いくつかのバージョンを持つことがある。<version>要素は、部品のバージョンの説明を考慮する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <version>

繰り返し要件: <version>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <version>要素は、LangString 要素として表現される。LangString は、50 キャラクターの SPM を持つ (詳細はセクション 4.2.11.2: LangString データタイプ参照)。

例:

```
<lom>
  <lifeCycle>
    <version>
      <string language="en">1.0 alpha</string>
    </version>
  </lifeCycle>
</lom>
```

コード例 4-14

4.2.3.2. <status> 要素

<status>要素は、SCORM コンテンツモデル部品の完了ステータスもしくは条件を記述する。部品は、ライフサイクル中にいくつかのステータスを持つことがある。(ドラフト、最終版など)。<status>要素は、部品のステータスの記述を考慮する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <status>

繰り返し要件: <status>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <status>要素は語彙要素として表現される。(詳細はセクション 4.2.11.3: 語彙データタイプ参照)。

語彙トークン: SCORM は、<status>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは:

- **draft:** ドラフト状態の部品(開発者に決定される)
- **final:** 最終版の部品(開発者に決定される)
- **revised:** 最後のバージョンから修正された部品
- **unavailable:** ステータス情報は入手可能でない

例：

```
<lom>
  <lifeCycle>
    <status>
      <source>LOMv1.0</source>
      <value>final</value>
    </status>
  </lifeCycle>
</lom>
```

コード例 4-15

4.2.3.3. <contribute>要素

<contribute>要素は、SCORM コンテンツモデル部品のライフサイクル中にその状態（作成，編集，訂正，出版など）に貢献した実態（人，団体など）を記述するのに利用される。貢献要素は，関与した全ての個人および団体を把握するのを可能にする。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <contribute>

繰り返し要件：<contribute>要素は，0 ないしそれ以上現れなくてはならない。<contribute>要素の SPM は 30 である。

データタイプ：<contribute>要素は，親要素である。親要素は，自身に関連付けられた値を持たない。親要素は，他の要素に対してコンテナとしての役割を果たす。<contribute>要素は，以下の子要素を含む：

- <role> - <contribute>要素が利用されると必須
- <entity> - <contribute>要素が利用されると必須
- <date> - <contribute>要素が利用されるとオプション

例：

```
<lom>
  <lifeCycle>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>author</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe FridayEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">A description for the date</string>
        </description>
      </date>
    </contribute>
  </lifeCycle>
</lom>
```

コード例 4-16

4.2.3.3.1. <role>要素

<role>要素は、貢献者になされた貢献（Entity 要素によって識別された）の種類を。IEEE は、部品のライフサイクルに関わる典型的な役割を定義した。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <role>

繰り返し要件: <role>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <role>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン: SCORM は、<role>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求しない。SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

- author
- publisher
- unknown
- initiator
- terminator
- validator
- editor
- graphical designer
- technical implementer
- content provider
- technical validator
- educational validator
- script writer
- instructional designer
- subject matter expert

例：

```
<lom>
  <lifeCycle>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>author</value>
      </role>
    </contribute>
  </lifeCycle>
</lom>
```

コード例 4-17

4.2.3.3.2. <entity>要素

<entity>要素は、SCORM コンテンツモデル部品の開発ライフサイクル中に貢献した実態を識別する。実態は、個人、団体などがある。一つ以上の実態を一覧にした場合、最も貢献度の高いものが最初に来る。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <entity>

繰り返し要件: <entity>要素は、0 ないしそれ以上現れなくてはならない。<entity>要素の SPM は 40 である。

データタイプ: <entity>要素は、文字列要素として表現される。文字列は、1000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.1: 文字列データタイプ参照)。全ての実態値は vCard [9] フォーマットで表される。これは、システムが<entity>要素によって表される文字列を取り込み、この文字列を有効な vCard として処理することを可能にする。

例:

```
<lom>
  <lifeCycle>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>author</value>
      </role>
      <entity>BEGIN:VCARD\r\nFN:Joe AuthorEND:VCARD</entity>
      <entity>BEGIN:VCARD\r\nFN:Mary AuthorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">A description for the date</string>
        </description>
      </date>
    </contribute>
  </lifeCycle>
</lom>
```

コード例 4-18

4.2.3.3.3. <date>要素

<date>要素は実態になされた貢献の日付を特定する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <date>

繰り返し要件: <date>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: 日付要素は、DateTime データタイプとして表される (詳細はセクション 4.2.11.4: DateTime データタイプ参照)。<date>要素は、親要素である。親要素は、自身に関連

付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<date>要素は、二つの要素を含む。一つは貢献の実際の日付(<dateTime>)を表し、もう一つは日付のテキスト記述を表す(<description>)：

- <dateTime>
- <description>

例：

```
<lom>
  <lifeCycle>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>author</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe AuthorEND:VCARD</entity>
      <entity>BEGIN:VCARD\nFN:Mary AuthorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">This date represents the date the author
finished authoring the component.</string>
        </description>
      </date>
    </contribute>
  </lifeCycle>
</lom>
```

コード例 4-19

4.2.4. <metaMetadata> 要素

Meta Metadata カテゴリは、SCORM コンテンツモデル部品レコードを記述する要素ではなく、メタデータレコード自身を記述する要素を提供する。このカテゴリは、どのようにメタデータインスタンス自身が識別されるか、誰がメタデータを作成したか、どのように、いつ、どういう参照をしたのかなどを記述する。

XML 名前空間: `http://ltsc.ieee.org/xsd/LOM`

XML 名前空間接頭辞: `lom`

XML バインディング表現: `<metaMetadata>`

繰り返し要件: <metaMetadata>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <metaMetadata>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<metaMetadata>要素は以下の子要素を含む:

- <identifier>
- <contribute>
- <metadataSchema>
- <language>

例:

```
<lom>
  <metaMetadata>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/metadata/MDO_01</entry>
    </identifier>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>creator</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe Metadata CreatorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">This date represents the date the creator
finished authoring the metadata.</string>
        </description>
      </date>
    </contribute>
    <metadataSchema>LOMv1.0</metadataSchema>
    <metadataSchema>SCORM_CAM_v1.3</metadataSchema>
    <language>en</language>
  </metaMetadata>
</lom>
```

コード例 4-20

4.2.4.1. <identifier>要素

<identifier>要素は、SCORM コンテンツモデル部品を記述するメタデータレコードを識別するグローバルに固有なラベルを割り当てるメカニズムを表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <identifier>

繰り返し要件: <identifier>要素は、0 ないしそれ以上現れなくてはならない。<identifier>要素の SPM は 10 である。

データタイプ: <identifier>要素は、親要素である。親要素は、自身に関連付けされた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<identifier>要素は、以下の子要素を含む:

- <catalog> - optional
- <entry> - mandatory

例:

```
<lom>
  <metaMetadata>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/metadata/MDO_01</entry>
    </identifier>
  </metaMetadata>
</lom>
```

コード例 4-21

4.2.4.1.1. <catalog> 要素

<catalog>要素は、エントリに対する識別名称もしくはカタログ化スキームを表す。様々な利用可能なカタログ化システムがある。SCORM はある特定のカタログ化システムの利用を要求しない。組織は、自分の組織の慣行および方針を満たすカタログ化スキームを自由に選択する。いくつかのタイプのカタログ化システムは:

- Universal Resource Identifier (URI)
- Universal Resource Name (URN)
- Digital Object Identifier (DOI)
- International Standard Book Numbers (ISBN)
- International Standard Serial Numbers (ISSN)

<catalog>要素は、エントリを作成し管理するのに利用されるスキームを表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <catalog>

繰り返し要件: <catalog>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <catalog>要素は、文字列要素として表現される。文字列は 1000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.1: 文字列データタイプ参照)。

例:

```
<lom>
  <metaMetadata>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/metadata/MD0_01</entry>
    </identifier>
  </metaMetadata>
</lom>
```

コード例 4-22

4.2.4.1.2. <entry>要素

<entry>要素は、メタデータを指定あるいは、識別するカタログ化スキーム内の識別子の値を表す。

識別子は、様々なフォーマットを取り入れることが可能である。IEEE は、実際の識別子の値が文字列として表されることを要求する。組織は、自分の組織の慣行および方針を満たす固有の識別子を作成するどんなメカニズムも自由に選択できる。

以下のリストは識別子値 (エントリ) の例である:

スキーム (<catalog>)	値 (<entry>)
Universal Resource Name	urn:ADL: 1345-GFGC-23ED-3321
Universal Resource Identifier	http://www.adlnet.org/content/C0_01
ADL Registry	2134-RF43-3233-FRI9-ACDA

ADL Note: ADL Registry は存在しない。これは、存在する可能性のある登録システムの一つのタイプの単なる例である。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <entry>

繰り返し要件: <entry>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <entry>要素の値は文字列要素として表現される。文字列は 1000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.1: 文字列データタイプ参照)。

例：

```
<lom>
  <metaMetadata>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.adlnet.org/metadata/MDO_01</entry>
    </identifier>
  </metaMetadata>
</lom>
```

コード例4-23

4.2.4.2. <contribute> 要素

<contribute>要素は、メタデータインスタンスの開発ライフサイクル中に、メタデータ(説明されてきている SCORM コンテンツモデル部品ではなく)インスタンスの状態に影響を与えた実態(人、団体など)を記述するのに利用される。貢献要素は、関与した全ての個人および団体を把握するのを可能にする。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <contribute>

繰り返し要件：<contribute>要素は、0 ないしそれ以上現れなくてはならない。<contribute>要素の SPM は 10 である。

データタイプ：<contribute>要素は、親要素である。親要素は自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<contribute>要素は、以下の子要素を含む：

- <role> - <contribute>要素が利用されると必須
- <entity> - <contribute>要素が利用されると必須
- <date> - <contribute>要素が利用されるとオプション

例：

```
<lom>
  <metaMetadata>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>creator</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe Metadata CreatorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">This date represents the date the creator
finished authoring the metadata.</string>
        </description>
      </date>
    </contribute>
  </metaMetadata>
</lom>
```

コード例 4-24

4.2.4.2.1. <role>要素

<role>要素は、貢献者によってなされた貢献の種類を定義する。IEEE は、メタデータインスタンスの開発ライフサイクルに関わる典型的な役割を定義した。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <role>

繰り返し要件: <role>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <role>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン: SCORM は、<role>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求しない。SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

- creator
- validator

例：

```

<lom>
  <metaMetadata>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>creator</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe Metadata CreatorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">This date represents the date the creator
finished authoring the metadata.</string>
        </description>
      </date>
    </contribute>
  </metaMetadata>
</lom>

```

コード例 4-25

4.2.4.2.2. <entity>要素

<entity>要素は、メタデータインスタンスの開発ライフサイクル中に貢献した実態を識別する。実態は、個人、団体などがある。一つ以上の実態が一覧にのる場合、最も貢献度の高いものが最初に来る。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <entity>

繰り返し要件: <entity>要素は、0 ないしそれ以上現れなくてはならない。<entity>要素の SPM は 10 である。

データタイプ: <entity>要素は、文字列要素として表現される。文字列は、1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。全ての実態値は、vCard [9] フォーマットで表される。これは、システムが<entity>要素によって表される文字列を取り込み、この文字列を有効な vCard として処理することを認める。

例：

```
<lom>
  <metaMetadata>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>creator</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe Metadata CreatorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">This date represents the date the creator
finished authoring the metadata.</string>
        </description>
      </date>
    </contribute>
  </metaMetadata>
</lom>
```

コード例 4-26

4.2.4.2.3. <date>要素

<date>要素は、実態になされた貢献の日付を識別する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <date>

繰り返し要件: <date>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: 日付要素は、DateTime データタイプとして表される(詳細はセクション 4.2.11.4: *DateTime* データタイプ参照)。<date>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<date>要素は、二つの要素を含む。一つは貢献の実際の日付(<dateTime>)を表し、もう一つは日付のテキスト記述を表す(<description>)：

- <dateTime>
- <description>

例：

```
<lom>
  <metaMetadata>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>creator</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe Metadata CreatorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">This date represents the date the creator
finished authoring the metadata.</string>
        </description>
      </date>
    </contribute>
  </metaMetadata>
</lom>
```

コード例 4-27

4.2.4.3. <metadataSchema>要素

<metadataSchema>要素は、メタデータインスタンスを作成するのに利用される承認された仕様の名称とバージョンを表す。複数の値が提供されると、メタデータインスタンスは、複数メタデータスキーマを遵守する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <metadataSchema>

繰り返し要件: <metadataSchema>要素は、0 ないしそれ以上現れなくてはならない。
<metadataSchema>要素の SPM は 10 である。

メタデータインスタンスが IEEE LOM XML バインディングに準拠するように生成され、<metadataSchema>要素が存在する場合、IEEE 標準規格はこの要素が標準規格に準拠していることを要求する。この場合、<metadataSchema>要素は、値 LOMv1.0 を持たなくてはならない。

データタイプ: <metadataSchema>要素は、文字列要素として表現される。文字列は、30 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。

例：

```
<lom>
  <metaMetadata>
    <!-- Mandatory Element/Values for all SCORM 1.3 Meta-data instances -->
    <metadataSchema>LOMv1.0</metadataSchema>
    <metadataSchema>SCORM_CAM_v1.3</metadataSchema>
  </metaMetadata>
</lom>
```

コード例 4-28

4.2.4.4. <language>要素

<language>要素は、メタデータインスタンスの言語を表す（LangStrings で見つけれられる全ての値の言語等々）。この値は、全ての LangStrings に対してデフォルト言語を表す。このデータ要素の値が、メタデータインスタンスになれば、LangString 値に対するデフォルト言語はない。この値が提供されると、LangString 要素に対して、言語値を示す必要はない。

XML 名前空間: `http://ltsc.ieee.org/xsd/LOM`

XML 名前空間接頭辞: `lom`

XML バインディング表現: `<language>`

繰り返し要件: <language>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <language>要素は、文字列要素として表現される。文字列は 100 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。

例:

```
<lom>
  <metaMetadata>
    <language>en</language>
  </metaMetadata>
</lom>
```

コード例 4-29

4.2.5. <technical>要素

Technical カテゴリは、SCORM コンテンツモデル部品の全てのテクニカルな特徴と要件を記述する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <technical>

繰り返し要件: <technical>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <technical>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<technical>要素は、以下の子要素を含む:

- <format>
- <size>
- <location>
- <requirement>
- <installationRemarks>
- <otherPlatformRequirements>
- <duration>

例：

```
<lom>
  <technical>
    <format>text/html</format>
    <size>1024</size>
    <location>Lesson01/Module01/Resources/SC001.htm</location>
    <requirement>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>ms-internet explorer</value>
        </name>
        <minimumVersion>5.0</minimumVersion>
        <maximumVersion>6.0</maximumVersion>
      </orComposite>
    </requirement>
    <installationRemarks>
      <string language="en">This activity requires the client browser to have
a Macromedia Flash plugin installed.</string>
    </installationRemarks>
    <otherPlatformRequirements>
      <string language="en">Sound card, Min. RAM: 16Mb, Video card and
display: at least 800 X 600 pixels x 256 colors</string>
    </otherPlatformRequirements>
    <duration>
      <duration>P5Y</duration>
      <description>
        <string language="en">Length of time to play simulation</string>
      </description>
    </duration>
  </technical>
</lom>
```

コード例 4-30

4.2.5.1. <format>要素

<format>要素は、SCORM コンテンツモデル部品の構成で利用される全ての部品のテクニカルなデータタイプを表す。この要素は、部品にアクセスするおよび部品を利用するソフトウェアニーズを識別するのに利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <format>

繰り返し要件: <format>要素は、0 ないしそれ以上現れなくてはならない。<format>要素の SPM は 40 である。

データタイプ: <format>要素は、文字列要素として表現される。文字列は、100 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。文字列は、IANA 登録 (RFC 2048:1996 参照) もしくはリテラル列（数字でない）に基づいて、MIME タイプである。

例：

```
<lom>
  <technical>
    <format>video/mpeg</format>
    <format>text/html</format>
  </technical>
</lom>
```

コード例 4-31

4.2.5.2. <size>要素

<size>要素は、デジタル SCORM コンテンツモデル部品のサイズをバイトで表す。サイズは 10 進法の値として表される（基数 10）。“0” から “9” が利用される。このデータは、SCORM 部品の実際のサイズを示す。部品が圧縮されると、このデータ要素は、圧縮されないサイズを示す。部品のサイズを決定するとき、以下がベストプラクティスとして提供される：

- コンテンツアグリゲーション： コンテンツパッケージのサイズ（コンテンツアグリゲーション等々）。パッケージが、PIF に圧縮されると、このサイズは、圧縮されたサイズを示す。パッケージが圧縮されないと、サイズはパッケージの全てのファイルのサイズを反映する（全てのファイルサイズの累計）
- コンテンツオーガニゼーション： コンテンツオーガニゼーションのサイズ。このサイズはコンテンツオーガニゼーション(コース、レッスンあるいは、コンテンツオーガニゼーションが表わすものなど)のサイズだけを反映すべきである。このサイズは、コンテンツオーガニゼーションで必ずしも反映されないパッケージが、ファイルに必要とされる（マニフェスト認証のために必要なコントロールファイル等々）ことがあるので、パッケージのサイズと違うことがある
- アクティビティ： アクティビティのサイズ（imsmanifest.xml ファイルで <organization>もしくは<item>要素として表現される）。このサイズは、アクティビティの構成に関与するファイルだけを反映する。アクティビティのサイズは、記述されたアクティビティが、他のアクティビティから成り立っているかどうか、もしくはアクティビティが、スタンドアロンのアクティビティかによる
- SCO： SCO のサイズ。このサイズは SCO を掲示しているリソースのサイズを反映する。これは、SCO の構成に利用されるサポーティングファイルの全てを含む
- Asset： 記述されているアセットのサイズ

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <size>

繰り返し要件： <size>要素は、0 ないし 1 回現れなくてはならない。

データタイプ： <size>要素は、文字列要素として表現される。文字列は、30 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。

例：

```
<lom>
  <technical>
    <size>345</size>
  </technical>
</lom>
```

コード例 4-32

4.2.5.3. <location>要素

<location>要素は、メタデータに記述された SCORM コンテンツモデル部品のロケーションを特定する列である。

<location>要素の利用は、パッケージのコンテンツの配信もしくは解体に対して要求されるロケーション情報が、マニフェストの<resource>および<file>要素に対する Href 属性で URL に提供されるので、SCORM メタデータでは推奨されない。

この要素は、コンテンツパッケージ自身に加えて、SCORM コンテンツ部品が見つかる一つ以上の代替ロケーションを記述するのに利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <location>

繰り返し要件: <location>要素は、0 ないしそれ以上現れなくてはならない。<location>要素の SPM は 10 である。

データタイプ: <location>要素は、文字列要素として表現される。文字列は、1000 キャラクタの SPM を持つ (詳細はセクション 4.2.11. 1: 文字列データタイプ参照)。

例：

```
<lom>
  <technical>
    <location>http://www.adlnet.org/content/Asset.jpg</location>
  </technical>
</lom>
```

コード例 4-33

4.2.5.4. <requirement> 要素

<requirement>要素は、SCORM コンテンツモデル部品の利用に対して必要なテクニカルな能力を表現する。<requirement>要素は、繰り返されることが可能である。複数要件が必要であれば、全ての要件が要求される (論理コネクターは AND である)。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <requirement>

繰り返し要件：<requirement>要素は、0 ないしそれ以上現れなくてはならない。<requirement>要素の SPM は 40 である。

データタイプ：<requirement>要素は、親要素である。親要素は自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<requirement>要素は、以下の子要素を含む：

- <orComposite>

例：

```
<lom>
  <technical>
    <requirement>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>ms-internet explorer</value>
        </name>
        <minimumVersion>5.0</minimumVersion>
        <maximumVersion>6.0</maximumVersion>
      </orComposite>
    </requirement>
  </technical>
</lom>
```

コード例 4-34

4.2.5.4.1. <orComposite>要素

<orComposite>要素は、単一要件を表す。複数の<orComposite>要素は論理コネクター “or” で接続される。

XML 名前空間： http://ltsc.ieee.org/xsd/LOM

XML 名前空間接頭辞： lom

XML バインディング表現： <orComposite>

繰り返し要件：<orComposite>要素は、0 ないしそれ以上現れなくてはならない。<orComposite>要素の SPM は 40 である。

データタイプ：<orComposite>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<orComposite>要素は以下の子要素を含む：

- <type>
- <name>
- <minimumVersion>
- <maximumVersion>

例： 以下のメタデータは、記述された部品が、以下の Web ブラウザのどれかで実行されることを示す：

- Microsoft Internet Explorer
 - o 最低バージョン：5.0
 - o 最高バージョン：6.0

もしくは

- Netscape Communicator
 - o 最低バージョン：4.7.9
 - o 最高バージョン：5.0

```
<lom>
  <technical>
    <requirement>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>ms-internet explorer</value>
        </name>
        <minimumVersion>5.0</minimumVersion>
        <maximumVersion>6.0</maximumVersion>
      </orComposite>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>netscape communicator</value>
        </name>
        <minimumVersion>4.7.9</minimumVersion>
        <maximumVersion>5.0</maximumVersion>
      </orComposite>
    </requirement>
  </technical>
</lom>
```

コード例 4-35

4.2.5.4.1.1. <type>要素

<type>要素は、SCORM コンテンツモデル部品を利用するために要求されるテクノロジーを表す（ハードウェア、ソフトウェア、ネットワークなど）。

メタデータインスタンスが<type>要素を含んだら、対応する<name>要素が、タイプに関するより詳細を記述するために、存在することが推奨されるベストプラクティスである。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <type>

繰り返し要件：<type>要素は、0 ないし 1 回現れなくてはならない。

データタイプ：<type>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン： SCORM は、<type>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求しない。SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

- operating system
- browser

例：

```
<lom>
  <technical>
    <requirement>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>ms-internet explorer</value>
        </name>
        <minimumVersion>5.0</minimumVersion>
        <maximumVersion>6.0</maximumVersion>
      </orComposite>
    </requirement>
  </technical>
</lom>
```

コード例 4-36

4.2.5.4.1.2. <name>要素

<name>要素は、SCORM コンテンツモデル部品を利用するために要求されるテクノロジーを表す。名前要素に利用される値は、値要素で識別された値による。

メタデータインスタンスが<name>要素を含んだら、対応する<type>要素が、要求されるテクノロジーに関するより詳細を記述するために、存在することが推奨されるベストプラクティスである。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <name>

繰り返し要件：<name>要素は、0 ないし 1 回現れなくてはならない。

データタイプ：<name>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン： SCORM は、<name>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求しない。SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

Requirement.OrComposite.Type = “operating system”であれば、

- pc-dos
- ms-windows
- macos
- unix
- multi-os
- none

Requirement.OrComposite.Type = “browser”であれば、

- any
- netscape-communicator
- ms-internet-explorer
- opera
- amaya

例：

```
<lom>
  <technical>
    <requirement>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>ms-internet-explorer</value>
        </name>
        <minimumVersion>5.0</minimumVersion>
        <maximumVersion>6.0</maximumVersion>
      </orComposite>
    </requirement>
  </technical>
</lom>
```

コード例4-37

4.2.5.4.1.3. <minimumVersion> 要素

<minimumVersion>要素は、SCORM コンテンツモデル部品を利用するために要求されるテクノロジーの最低バージョンを表す。最低バージョンが提示する要求されるテクノロジーは<name> および<value>要素によって記述される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <minimumVersion>

繰り返し要件: <minimumVersion>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <minimumVersion>要素は、文字列要素として表現される。文字列は、30 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。

例：

```
<lom>
  <technical>
    <requirement>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>ms-internet explorer</value>
        </name>
        <minimumVersion>5.0</minimumVersion>
        <maximumVersion>6.0</maximumVersion>
      </orComposite>
    </requirement>
  </technical>
</lom>
```

コード例 4-38

4.2.5.4.1.4. <maximumVersion>要素

<maximumVersion>要素は、SCORM コンテンツモデル部品を利用するために要求されるテクノロジーの最高バージョンを表す。最高バージョンが提示する要求されるテクノロジーは、<name> および<value>要素によって記述される。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <maximumVersion>

繰り返し要件： <maximumVersion>要素は、0 ないし 1 回現れなくてはならない。

データタイプ： <maximumVersion>要素は、文字列要素として表現される。文字列は、30 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。

例：

```
<lom>
  <technical>
    <requirement>
      <orComposite>
        <type>
          <source>LOMv1.0</source>
          <value>browser</value>
        </type>
        <name>
          <source>LOMv1.0</source>
          <value>ms-internet explorer</value>
        </name>
        <minimumVersion>5.0</minimumVersion>
        <maximumVersion>6.0</maximumVersion>
      </orComposite>
    </requirement>
  </technical>
</lom>
```

コード例 4-39

4.2.5.5. <installationRemarks>要素

<installationRemarks>要素は、SCORM コンテンツモデル部品をどのようにインストールするかについての特定の指示を表すにも利用される。この要素は、部品のユーザ（LMS、コンテンツ開発者、編集ツールなど）にある特定の指示を説明するのに利用される。SCORM 部品のテクニカル要件をより詳細に記述する表わすのに利用されることもある。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <installationRemarks>

繰り返し要件: <installationRemarks>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <installationRemarks>要素は、LangString として表される。LangString は、1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11.2: *LangString* データタイプ参照）。

例：

```
<lom>
  <technical>
    <installationRemarks>
      <string language="en">This activity requires the client browser to have
a Macromedia Flash plugin installed.</string>
    </installationRemarks>
  </technical>
</lom>
```

コード例 4-40

4.2.5.6. <otherPlatformRequirements>要素

<otherPlatformRequirements>要素は、SCORM コンテンツモデル部品の他のソフトウェア要件およびハードウェア要件に関する情報を表す。この要素は、他のテクニカル要素で表現できない要件を記述するのに利用されるべきである。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <otherPlatformRequirements>

繰り返し要件: <otherPlatformRequirements>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <otherPlatformRequirements>要素は、LangString として表される。LangString は、1000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.2: *LangString* データタイプ参照)。

例:

```
<lom>
  <technical>
    <otherPlatformRequirements>
      <string language="en">Sound card, Min. RAM: 16Mb, Video card and
display: at least 800 X 600 pixels x 256 colors</string>
    </otherPlatformRequirements>
  </technical>
</lom>
```

コード例 4-41

4.2.5.7. <duration>要素

<duration>要素は、継続する SCORM コンテンツモデル部品が、意図されたスピードで実行されたときにかける時間を表す。この要素は、音、映画、シミュレーションなどに役に立つ。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <duration>

繰り返し要件: <duration>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <duration>要素は、Duration データタイプとして表される。(詳細はセクション 4.2.11. 5: *Duration* データタイプ参照)。

例：

```
<lom>
  <technical>
    <duration>
      <!-- Movie will play for 1 hour and 30 minutes -->
      <duration>PT1H30M</duration>
      <description>
        <string language="en">Length of time to play movie</string>
      </description>
    </duration>
  </technical>
</lom>
```

コード例 4-42

4.2.6. <educational> 要素

Educational カテゴリは、SCORM コンテンツモデル部品の主な教育的もしくは学術的な特徴を記述する。このカテゴリは、教育的な特徴の説明を考慮し、典型的には、教師、管理者、編集者および学習者に利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <educational>

繰り返し要件: <educational>要素は、0 ないしそれ以上現れなくてはならない。<educational>要素の SPM は 100 である。

データタイプ: <educational>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<educational>要素は、以下の子要素を含む：

- <interactivityType>
- <learningResourceType>
- <interactivityLevel>
- <semanticDensity>
- <intendedEndUserRole>
- <context>
- <typicalAgeRange>
- <difficulty>
- <typicalLearningTime>
- <description>
- <language>

例：

```
<lom>
  <educational>
    <interactivityType>
      <source>LOMv1.0</source>
      <value>mixed</value>
    </interactivityType>
    <learningResourceType>
      <source>LOMv1.0</source>
      <value>figure</value>
    </learningResourceType>
    <learningResourceType>
      <source>LOMv1.0</source>
      <value>narrative text</value>
    </learningResourceType>
    <interactivityLevel>
      <source>LOMv1.0</source>
      <value>very low</value>
    </interactivityLevel>
    <semanticDensity>
      <source>LOMv1.0</source>
      <value>very low</value>
    </semanticDensity>
    <intendedEndUserRole>
      <source>LOMv1.0</source>
      <value>learner</value>
    </intendedEndUserRole>
    <context>
      <source>LOMv1.0</source>
      <value>training</value>
    </context>
    <typicalAgeRange>
      <string language="en">18-</string>
    </typicalAgeRange>
    <difficulty>
      <source>LOMv1.0</source>
      <value>easy</value>
    </difficulty>
    <typicalLearningTime>
      <duration>PT1H30M</duration>
      <description>
        <string language="en">Average length of time to experience the
activity.</string>
      </description>
    </typicalLearningTime>
    <language>en-US</language>
  </educational>
</lom>
```

コード例 4-43

4.2.6.1. <interactivityType> 要素

<interactivityType>要素は、SCORM コンテンツモデル要素にサポートされる学習の支配的なモードを表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <interactivityType>

繰り返し要件: <interactivityType>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <interactivityType>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン: SCORM は、<interactivityType>要素を、制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1 -2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- **active:** 対話型学習（能動的学習）は、学習者による生産的なアクションを直接生み出すコンテンツにサポートされる
- **expositive:** 掲示型学習（受動的学習）は、学習者の行うことが、主に提示されたコンテンツを吸収することから成り立つときに発生する
- **mixed:** 対話型と掲示型タイプの組み合わせ

例:

```
<lom>
  <educational>
    <interactivityType>
      <source>LOMv1.0</source>
      <value>mixed</value>
    </interactivityType>
  </educational>
</lom>
```

コード例 4-44

4.2.6.2. <learningResourceType> 要素

<learningResourceType>要素は、特定の種類の SCORM コンテンツモデル部品を表す。この要素は、部品で利用されたリソースのタイプを完全に記述するために、繰り返されることが可能である。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <learningResourceType>

繰り返し要件: <learningResourceType>要素は、0 ないしそれ以上現れなくてはならない。

<learningResourceType>要素の SPM は 10 である。

データタイプ: <learningResourceType>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン: SCORM は、<learningResourceType>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1 -2002 で定義された語彙の利用を要求しない。SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

- **exercise**
- **simulation**
- **questionnaire**
- **diagram**

-
- figure
 - graph
 - index
 - slide
 - table
 - narrative text
 - exam
 - experiment
 - problem statement
 - self assessment
 - lecture

例：

```
<lom>
  <educational>
    <learningResourceType>
      <source>LOMv1.0</source>
      <value>narrative text</value>
    </learningResourceType>
    <learningResourceType>
      <source>LOMv1.0</source>
      <value>simulation</value>
    </learningResourceType>
  </educational>
</lom>
```

コード例 4-45

4.2.6.3. <interactivityLevel>要素

<interactivityLevel>は、SCORM コンテンツモデル部品を特徴付ける相互作用性のレベルを示す。この文脈での相互作用性は、学習者が部品のある側面もしくは動作に影響を与えるレベルを示す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <interactivityLevel>

繰り返し要件: <interactivityLevel>要素は、0 ないし 1 回現れなくてはならない。データタイプ: <interactivityLevel>要素は語彙要素として表現される。(詳細はセクション 4.2.11.3: 語彙データタイプ参照)。

語彙トークン: SCORM は、<interactivityLevel>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- very low
- low
- medium
- high
- very high

これらの値は、本質的に、コミュニティの慣習の文脈の中でのみ意味をもつ。現時点で、ADL コミュニティは、この尺度を定義していない。ADL コミュニティが、この尺度を定義しようとしたら、この情報は ADL テクニカルチームで取り上げられる。現時点でこの尺度の定義は組織次第である。

例：

```
<lom>
  <educational>
    <interactivityLevel>
      <source>LOMv1.0</source>
      <value>very low</value>
    </interactivityLevel>
  </educational>
</lom>
```

コード例 4-46

4.2.6.4. <semanticDensity>要素

<semanticDensity>は、SCORM コンテンツモデル部品の簡潔さの度合を示す。SCORM 部品のセマンティック密度は、そのサイズ、範囲、もしくはオーディオやビデオといった自分で時間計測されるリソースの場合は、期間といったかたちで予測される。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <semanticDensity>

繰り返し要件： <semanticDensity>要素は、0 ないし 1 回現れなくてはならない。

データタイプ： <semanticDensity>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン： SCORM は、<semanticDensity>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1 2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- very low
- low
- medium
- high
- very high

これらの値は、本質的に、コミュニティの慣習の文脈の中でのみ意味をもつ。現時点で、ADL コミュニティはこの尺度を定義していない。ADL コミュニティがこの尺度を定義しようとしたら、この情報は ADL テクニカルチームで取り上げられる。現時点でこの尺度の定義は組織次第である。

例：

```
<lom>
  <educational>
    <semanticDensity>
      <source>LOMv1.0</source>
      <value>very low</value>
    </semanticDensity>
  </educational>
</lom>
```

コード例 4-47

4.2.6.5. <intendedEndUserRole> 要素

<intendedEndUserRole>要素は、SCORM コンテンツモデル要素の設計の対象とされた主なユーザを表す。複数の要素が利用されたら、最も支配的な役割が最初に来る。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <intendedEndUserRole>

繰り返し要件: <intendedEndUserRole>要素は、0 ないしそれ以上現れなくてはならない。
<intendedEndUserRole>要素の SPM は 10 である。

データタイプ: <intendedEndUserRole>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン: SCORM は、<intendedEndUserRole>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1 2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- teacher
- author
- learner
- manager

例：

```
<lom>
  <educational>
    <intendedEndUserRole>
      <source>LOMv1.0</source>
      <value>learner</value>
    </intendedEndUserRole>
  </educational>
</lom>
```

コード例 4-48

4.2.6.6. <context>要素

<context>要素は、SCORM コンテンツモデル部品の学習と利用が起こるように意図された主な環境を表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <context>

繰り返し要件: <context>要素は、0 ないしそれ以上現れなくてはならない。<context>要素の SPM は 10 である。

データタイプ: <context>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: *語彙データタイプ*参照）。

語彙トークン: SCORM は、<context>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1 2002 で定義された語彙の利用を要求しない。SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

- school
- higher education
- training
- other

例:

```
<lom>
  <educational>
    <context>
      <source>LOMv1.0</source>
      <value>training</value>
    </context>
  </educational>
</lom>
```

コード例 4-49

4.2.6.7. <typicalAgeRange>要素

<typicalAgeRange>要素は、典型的なエンドユーザの年齢を表す。この要素は、暦年齢と発育年齢が違う場合、発育年齢を指す。IEEE 規格は、適用可能な場合、値は**最低年齢– 最高年齢**もしくは **最低年齢 – (18 – 25 もしくは 18 等)**とフォーマットされることを推奨する。この要素の値は、必ずしも数値である必要はない(大人のみ、7 歳以上の子に適切など)。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <typicalAgeRange>

繰り返し要件: <typicalAgeRange>要素は、0 ないしそれ以上現れなくてはならない。<typicalAgeRange>要素の SPM は 5 である。

データタイプ: <typicalAgeRange>要素は LangString として表される。LangString は 1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11.2: *LangString データタイプ*参照）。

例：

```
<lom>
  <educational>
    <typicalAgeRange>
      <string language="en">18-</string>
    </typicalAgeRange>
  </educational>
</lom>
```

コード例 4-50

4.2.6.8. <difficulty>要素

<difficulty>要素は、典型的な意図された対象者に対して、どの程度 SCORM コンテンツモデル部品が難しいかを表す。典型的な対象者は<context>および<typicalAgeRange>要素に特性を与えられる。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <difficulty>

繰り返し要件: <difficulty>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <difficulty>要素は、語彙要素として表現される。(詳細はセクション 4.2.11.3: 語彙データタイプ参照)。

語彙トークン: SCORM は、<difficulty>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- very low
- low
- medium
- high
- very high

これらの値は、本質的に、コミュニティの慣習の文脈の中でのみ意味をもつ。現時点で、ADL コミュニティは、この尺度を定義していない。ADL コミュニティが、この尺度を定義しようとしたら、この情報は ADL テクニカルチームで取り上げられる。現時点でこの尺度の定義は組織次第である。

例：

```
<lom>
  <educational>
    <difficulty>
      <source>LOMv1.0</source>
      <value>easy</value>
    </difficulty>
  </educational>
</lom>
```

コード例 4-51

4.2.6.9. <typicalLearningTime>要素

<typicalLearningTime>要素は、典型的な意図された対象者に対して、SCORM コンテンツモデル部品を通じて作業するのに要する典型的な時間の目安を示す。典型的な対象者は<context>および<typicalAgeRange>要素によって特徴付けられる。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <typicalLearningTime>

繰り返し要件: <typicalLearningTime>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <typicalLearningTime>要素は、Duration データタイプとして表される。（詳細はセクション 4.2.11. 5: *Duration* データタイプ参照）。

例:

```
<lom>
  <educational>
    <typicalLearningTime>
      <duration>PT1H30M</duration>
      <description>
        <string language="en">Average length of time to experience the
activity.</string>
      </description>
    </typicalLearningTime>
  </educational>
</lom>
```

コード例 4-52

4.2.6.10. <description>要素

<description>要素は、SCORM コンテンツ部品がどのように利用されるかに関するコメントをするのに利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <description>

繰り返し要件: <description>要素は、0 ないしそれ以上現れなくてはならない。<description>要素の SPM は 10 である。

データタイプ: <description>要素は、LangString として表される。LangString は、1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11.2: *LangString* データタイプ参照）。

例：

```
<lom>
  <educational>
    <description>
      <string language="en">This course is designed for IT professionals
responsible for implementing Java</string>
    </description>
  </educational>
</lom>
```

コード例 4-53

4.2.6.11. <language>要素

<language>要素は、SCORM コンテンツモデル部品の典型的な意図されたユーザに利用される人言語を表す。典型的な意図されたユーザは、要素<context>および<typicalAgeRange>によって特徴づけられる。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <language>

繰り返し要件： <language>要素は、0 ないしそれ以上現れなくてはならない。<language>要素の SPM は 10 である。

データタイプ： <language>要素の値は、文字列要素として表現される。文字列は、100 キャラクターの SPM を持つ（詳細はセクション 4.2.11.1: 文字列データタイプ参照）。

例：

```
<lom>
  <educational>
    <language>en-US</language>
  </educational>
</lom>
```

コード例 4-54

4.2.7. <rights>要素

Rights カテゴリは、SCORM コンテンツモデル部品の知的所有権および利用条件を記述する。この要素は、SCORM 部品の全てのデジタル著作権を記述する(利用コスト、コピーライトなど)利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <rights>

繰り返し要件: <rights>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <rights>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<rights>要素は、以下の子要素を含む:

- <cost>
- <copyrightAndOtherRestrictions>
- <description>

例:

```
<lom>
  <rights>
    <cost>
      <source>LOMv1.0</source>
      <value>yes</value>
    </cost>
    <copyrightAndOtherRestrictions>
      <source>LOMv1.0</source>
      <value>yes</value>
    </copyrightAndOtherRestrictions>
    <description>
      <string language="en">For additional information or questions regarding
copyright, distribution and reproduction, contact Joe Developer at
joe_developer@someorganization.org</string>
    </description>
  </rights>
</lom>
```

コード例 4-55

4.2.7.1. <cost>要素

<cost>要素は、SCORM コンテンツモデル部品が何らかの支払の種類を要求するかどうかを表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <cost>

繰り返し要件: <cost>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <cost>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン: SCORM は、<cost>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- yes
- no

<cost>要素が yes に設定されると、<description>要素が、コストに関する詳細を追加するのに利用されることが可能である。

例：

```
<lom>
  <rights>
    <cost>
      <source>LOMv1.0</source>
      <value>yes</value>
    </cost>
    <description>
      <string language="en">Contact joe_developer@someorg.org for cost
information.</string>
    </description>
  </rights>
</lom>
```

コード例 4-56

4.2.7.2. <copyrightAndOtherRestrictions>要素

<copyrightAndOtherRestrictions>要素は、著作権か他の制約が、SCORM コンテンツモデル部品の利用に適用するかどうかを記述する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <copyrightAndOtherRestrictions>

繰り返し要件: <copyrightAndOtherRestrictions>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <copyrightAndOtherRestrictions>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン: SCORM は、<copyrightAndOtherRestrictions>要素を制限された語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求する。IEEE で定義された有効なトークンは：

- yes
- no

<copyrightAndOtherRestrictions>要素が yes に設定されると、<description>要素がコストに関する詳細を追加するのに利用されることが可能である。

例：

```
<lom>
  <rights>
    <copyrightAndOtherRestrictions>
      <source>LOMv1.0</source>
      <value>yes</value>
    </copyrightAndOtherRestrictions>
    <description>
      <string language="en">Contact joe_developer@someorg.org for copyright
information.</string>
    </description>
  </rights>
</lom>
```

コード例 4-57

4.2.7.3. <description>要素

<description>要素は、SCORM コンテンツモデル部品の利用条件に関するコメントに備える。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <description>

繰り返し要件: <description>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <description>要素は LangString として表される。LangString は、1000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.2: *LangString* データタイプ参照)。

例：

```
<lom>
  <rights>
    <copyrightAndOtherRestrictions>
      <source>LOMv1.0</source>
      <value>yes</value>
    </copyrightAndOtherRestrictions>
    <description>
      <string language="en">For additional information or questions regarding
copyright, distribution and reproduction, contact Joe Developer at
joe_developer@someorganization.org</string>
    </description>
  </rights>
</lom>
```

コード例 4-58

4.2.8. <relation>要素

Relation カテゴリは、SCORM コンテンツモデル部品と、もしあれば、他の部品の関係を定義する。関係要素は、繰り返されることが認められている。複数の関係を定義するのに、このカテゴリのいくつかのインスタンスを作成することが可能である。一つ以上の対象部品がある場合、各対象は、新たな関係インスタンスを持つ。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <relation>

繰り返し要件: <relation>要素は、0 ないしそれ以上現れなくてはならない。<relation>要素の SPM は 100 である。

データタイプ: <relation>要素は、親要素である。親要素は自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<relation>要素は、以下の子要素を含む:

- <kind>
- <resource>

例:

```
<lom>
  <relation>
    <kind>
      <source>LOMv1.0</source>
      <value>isbasedon</value>
    </kind>
    <resource>
      <identifier>
        <catalog>URN</catalog>
        <entry>urn:ADL:1234-45FD</entry>
      </identifier>
      <description>
        <string language="en">Microsoft MSCE</string>
      </description>
    </resource>
  </relation>
</lom>
```

コード例 4-59

4.2.8.1. <kind>要素

<kind>要素は、SCORM コンテンツモデル部品とリソースによって識別された対象部品との関係性の特性を記述する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <kind>

繰り返し要件：<kind>要素は、0 ないし 1 回現れなくてはならない。

データタイプ：<kind>要素は、語彙要素として表現される。（詳細はセクション 4.2.11.3: 語彙データタイプ参照）。

語彙トークン：SCORM は、<kind>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求しない。しかしながら、SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

- ispartof
- haspart
- isversionof
- hasversion
- isformatof
- hasformat
- references
- isreferencedby
- isbasedon
- isbasisfor
- requires
- isrequiredby

例：

```
<lom>
  <relation>
    <kind>
      <source>LOMv1.0</source>
      <value>ispartof</value>
    </kind>
    <resource>
      <identifier>
        <catalog>URN</catalog>
        <entry>urn:ADL:1234-45FD-3324</entry>
      </identifier>
      <description>
        <string language="en">ADL Course: Microsoft MSCE</string>
      </description>
    </resource>
  </relation>
</lom>
```

コード例 4-60

4.2.8.2. <resource>要素

<resource>要素は、この関係性が参照する対象 SCORM コンテンツモデル部品を記述する。

XML 名前空間： <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞： lom

XML バインディング表現： <resource>

繰り返し要件：<resource>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <resource>要素は、親要素である。親要素は自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<resource>要素は、以下の子要素を含む：

- <identifier>
- <description>

例：

```
<lom>
  <relation>
    <kind>
      <source>LOMv1.0</source>
      <value>isbasedon</value>
    </kind>
    <resource>
      <identifier>
        <catalog>URN</catalog>
        <entry>urn:ADL:1234-45FD</entry>
      </identifier>
      <description>
        <string language="en">ADL Course: Microsoft MSCE</string>
      </description>
    </resource>
  </relation>
</lom>
```

コード例 4-61

4.2.8.2.1. <identifier>要素

<identifier>要素は、対象 SCORM コンテンツモデル部品を識別するグローバルに固有なラベルを割り当てるメカニズムを表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <identifier>

繰り返し要件: <identifier>要素は、0 ないしそれ以上現れなくてはならない。<identifier>要素の SPM は 10 である。

データタイプ: <identifier>要素は、親要素である。親要素は自身に関連づけられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<identifier>要素は以下の子要素を含む：

- <catalog> - 識別子が定義された場合オプション
- <entry> - 識別子が定義された場合必須

<identifier>要素が、メタデータインスタンスで利用されると、SCORM は、<entry>要素がメタデータインスタンスにあるように要求する。<entry>要素は、固有の識別子を保持する要素である。<catalog>要素は、オプションであるが、エントリに対して、カタログもしくは識別システムスキームを供給するのがベストプラクティスであるとみなされる。

例：

```
<lom>
  <relation>
    <kind>
      <source>LOMv1.0</source>
      <value>isbasedon</value>
    </kind>
    <resource>
      <identifier>
        <catalog>URN</catalog>
        <entry>urn:ADL:1234-45FD</entry>
      </identifier>
      <description>
        <string language="en">ADL Course: Microsoft MSCE</string>
      </description>
    </resource>
  </relation>
</lom>
```

コード例 4-62

4.2.8.2.1.1. <catalog>要素

<catalog>要素は、エントリに対する識別名称もしくはカタログ化スキームを表す。様々な利用可能なカタログ化システムがある。SCORM は、ある特定のカタログ化システムの利用を要求しない。組織は、自分の組織の慣行および方針を満たすカタログ化スキームを自由に選択する。いくつかのタイプのカタログ化システムは：

- Universal Resource Identifier (URI)
- Universal Resource Name (URN)
- Digital Object Identifier (DOI)
- International Standard Book Numbers (ISBN)
- International Standard Serial Numbers (ISSN)

<catalog>要素は エントリを作成し管理するスキームを表す。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <catalog>

繰り返し要件: <catalog>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <catalog>要素は、文字列要素として表現される。文字列は、1000 キャラクタの SPM を持つ (詳細はセクション 4.2.11.1: 文字列データタイプ参照)。

例：

```

<lom>
  <relation>
    <kind>
      <source>LOMv1.0</source>
      <value>isbasedon</value>
    </kind>
    <resource>
      <identifier>
        <catalog>URN</catalog>
        <entry>urn:ADL:1234-45FD</entry>
      </identifier>
      <description>
        <string language="en">ADL Course: Microsoft MSCE</string>
      </description>
    </resource>
  </relation>
</lom>

```

コード例 4-63

4.2.8.2.1.2. <entry>要素

<entry>要素は、対象 SCORM コンテンツモデル部品を識別するカタログ化スキーム内の識別子の値を表す。

識別子は、様々なフォーマットを取り入れることが可能である。IEEE は、実際の識別子の値が文字列として表されることを要求する。組織は、自分の組織の慣行および方針を満たす固有の識別子を作成するどんなメカニズムも自由に選択する。共通スキームが選択されることが推奨される。

以下の一覧は、識別子値（エントリ）の例である：

スキーム (<catalog>)	値 (<entry>)
Universal Resource Name	urn:ADL: 1345-GFGC-23ED-3321
Universal Resource Identifier	http://www.adlnet.org/content/C0_01
ADL Registry	2134-RF43-3233-FRI9-ACDA

ADL Note: ADL Registry は、存在しない。これは、存在する可能性のある登録システムの一つのタイプの単なる例である。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <entry>

繰り返し要件: <entry>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <entry>要素の値は、文字列要素として表現される。文字列は、1000 キャラクタの SPM を持つ（詳細はセクション 4.2.11.1: 文字列データタイプ参照）。

例:


```

<lom>
  <relation>
    <kind>
      <source>LOMv1.0</source>
      <value>isbasedon</value>
    </kind>
    <resource>
      <identifier>
        <catalog>URN</catalog>
        <entry>urn:ADL:1234-45FD</entry>
      </identifier>
      <description>
        <string language="en">ADL Course: Microsoft MSCE</string>
      </description>
    </resource>
  </relation>
</lom>

```

コード例 4-64

4.2.8.2.2. <description>要素

<description>要素は、対象 SCORM コンテンツ部品を記述する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <description>

繰り返し要件: <description>要素は、0 ないしそれ以上現れなくてはならない。<description>要素の SPM は 10 である。

データタイプ: <description>要素は、LangString として表される。LangString は、1000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.2: *LangString* データタイプ参照)。

例:

```

<lom>
  <relation>
    <kind>
      <source>LOMv1.0</source>
      <value>isbasedon</value>
    </kind>
    <resource>
      <identifier>
        <catalog>URN</catalog>
        <entry>urn:ADL:1234-45FD</entry>
      </identifier>
      <description>
        <string language="en">ADL Course: Microsoft MSCE</string>
      </description>
    </resource>
  </relation>
</lom>

```

コード例 4-65

4.2.9. <annotation>要素

Annotation カテゴリは、SCORM コンテンツモデル部品の教育的利用に関するコメントおよびいつ誰がそのコメントを作成したかに関する情報を提供する。このカテゴリは、教育者が SCORM コンテンツモデル部品の評価、利用に関する提案などを共有することを可能にする。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <annotation>

繰り返し要件: <annotation>要素は、0 ないしそれ以上現れなくてはならない。<annotation>要素の SPM は 30 である。

データタイプ: <annotation>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<annotation>要素は、以下の子要素を含む:

- <entity>
- <date>
- <description>

例:

```
<lom>
  <annotation>
    <entity>BEGIN:VCARD\nFN:Joe AuthorEND:VCARD</entity>
    <date>
      <dateTime>2001-07-30T10:14:35.5+01:00</dateTime>
      <description>
        <string language="en">Date and time annotation was created</string>
      </description>
    </date>
    <description>Learners will need to understand the fundamentals of Windows
programming in order to grasp the concepts described in this
learning.</description>
  </annotation>
</lom>
```

コード例 4-66

4.2.9.1. <entity>要素

<entity>要素は、注釈を作成する実態を識別する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <entity>

繰り返し要件: <entity>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <entity>要素は、文字列要素として表現される。文字列は、1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11. 1: 文字列データタイプ参照）。全ての実態値は、vCard [9] フォーマットで表される。これは、システムが<entity>要素によって表される文字列を取り込み、この文字列を有効な vCard として処理することを可能にする。

例:

```
<lom>
  <annotation>
    <entity>BEGIN:VCARD\nFN:Joe AuthorEND:VCARD</entity>
    <date>
      <dateTime>2001-07-30T10:14:35.5+01:00</dateTime>
      <description>
        <string language="en">Date and time annotation was created</string>
      </description>
    </date>
    <description>Learners will need to understand the fundamentals of Windows
programming in order to grasp the concepts described in this
learning.</description>
  </annotation>
</lom>
```

コード例 4-67

4.2.9.2. <date>要素

<date>要素は、注釈が作成された日付を識別する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <date>

繰り返し要件: <date>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: 日付要素は、DateTime データタイプとして表される（詳細はセクション 4.2.11.4: *DateTime* データタイプ参照）。<date>要素は、親要素である。親要素は、自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<date>要素は、二つの要素を含む。一つは貢献の実際の日付(<dateTime>)を表し、もう一つは日付のテキスト記述を表す(<description>)：

- <dateTime>
- <description>

例：

```
<lom>
  <annotation>
    <entity>BEGIN:VCARD\nFN:Joe AuthorEND:VCARD</entity>
    <date>
      <dateTime>2001-07-30T10:14:35.5+01:00</dateTime>
      <description>
        <string language="en">Date and time annotation was created</string>
      </description>
    </date>
    <description>Learners will need to understand the fundamentals of Windows
programming in order to grasp the concepts described in this
learning.</description>
  </annotation>
</lom>
```

コード例 4-68

4.2.9.3. <description>要素

<description>要素は、注釈の内容を表すために利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <description>

繰り返し要件: <description>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <description>要素は、LangString として表される。LangString は 1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11.2: *LangString* データタイプ参照）。

例：

```
<lom>
  <annotation>
    <entity>BEGIN:VCARD\nFN:Joe AuthorEND:VCARD</entity>
    <date>
      <dateTime>2001-07-30T10:14:35.5+01:00</dateTime>
      <description>
        <string language="en">Date and time annotation was created</string>
      </description>
    </date>
    <description>Learners will need to understand the fundamentals of Windows
programming in order to grasp the concepts described in this
learning.</description>
  </annotation>
</lom>
```

コード例 4-69

4.2.10. <classification>要素

Classification カテゴリは、ある特定の分類システムで、どこに SCORM コンテンツモデル部品があてはまるかを記述する。複数の分類を定義するのに、複数の Classification カテゴリが利用されることがある。Classification カテゴリは、一般的には、コントロールされた語彙もしくは分類システムへのリンクとして利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <classification>

繰り返し要件: <classification>要素は、0 ないしそれ以上現れなくてはならない。
<classification>要素の SPM は 40 である。

データタイプ: <classification>要素は、親要素である。親要素は自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<classification>要素は、以下の子要素を含む：

- <purpose>
- <taxonPath>
- <description>
- <keyword>

例：

```
<lom>
  <classification>
    <purpose>
      <source>LOMv1.0</source>
      <value>skill level</value>
    </purpose>
    <taxonPath>
      <source>
        <string language="en-US">ADL SCORM Concepts</string>
      </source>
      <taxon>
        <id>I</id>
        <entry>
          <string language="en-US">Content Aggregation Model</string>
        </entry>
      </taxon>
      <taxon>
        <id>I.A</id>
        <entry>
          <string language="en-US">Content Packaging Fundamentals</string>
        </entry>
      </taxon>
      <taxon>
        <id>I.A.3</id>
        <entry>
          <string language="en-US">Resource Fundamentals</string>
        </entry>
      </taxon>
      <taxon>
        <id>I.A.3.a</id>
        <entry>
          <string language="en-US">Packaging SCOs</string>
        </entry>
      </taxon>
    </taxonPath>
    <description>
      <string language="en-US">Describing and packaging SCOs in a SCORM
Content Package</string>
    </description>
    <keyword>
      <string language="en-US">Packaging SCOs</string>
    </keyword>
  </classification>
</lom>
```

コード例 4-70

4.2.10.1. <purpose>要素

<purpose>要素は、SCORM コンテンツモデル部品を分類する目的を定義する。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <purpose>

繰り返し要件: <purpose>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <purpose>要素は、語彙要素として表現される。(詳細はセクション 4.2.11.3: 語彙データタイプ参照)。

語彙トークン: SCORM は、<purpose>要素をベストプラクティス語彙要素として定義する。SCORM は、IEEE 1484.12.1-2002 で定義された語彙の利用を要求しない。SCORM は、ベストプラクティスとして値の利用を推奨する。IEEE で定義された有効なトークンは：

- discipline
- idea
- prerequisite
- educational objective
- accessibility restrictions
- educational level
- skill level
- security level
- competency

例：

```
<lom>
  <classification>
    <purpose>
      <source>LOMv1.0</source>
      <value>skill level</value>
    </purpose>
  </classification>
</lom>
```

コード例 4-71

4.2.10.2. <taxonPath>要素

<taxonPath>要素は、ある特定の分類システムの分類ルートを記述する。各々の後続レベルは、前のレベルの定義をより細かく分ける。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <taxonPath>

繰り返し要件: <taxonPath>要素は、0 ないしそれ以上現れなくてはならない。<taxonPath>要素の SPM は 15 である。

データタイプ: <taxonPath>要素は、親要素である。親要素は自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<taxonPath>要素は、以下の子要素を含む：

- <source>
- <taxon>

例：

```
<lom>
  <classification>
    <purpose>
      <source>LOMv1.0</source>
      <value>skill level</value>
    </purpose>
    <taxonPath>
      <source>
        <string language="en-US">ADL SCORM Concepts</string>
      </source>
      <taxon>
        <id>I</id>
        <entry>
          <string language="en-US">Content Aggregation Model</string>
        </entry>
      </taxon>
      <taxon>
        <id>I.A</id>
        <entry>
          <string language="en-US">Content Packaging Fundamentals</string>
        </entry>
      </taxon>
      <taxon>
        <id>I.A.3</id>
        <entry>
          <string language="en-US">Resource Fundamentals</string>
        </entry>
      </taxon>
      <taxon>
        <id>I.A.3.a</id>
        <entry>
          <string language="en-US">Packaging SCOs</string>
        </entry>
      </taxon>
    </taxonPath>
  </classification>
</lom>
```

コード例 4-72

4.2.10.2.1. <source>要素

<source>要素は、分類システムを記述し、もしくは命名する。このデータ要素は、認識された公式のどんな分類も、もしくはどんなユーザー定義の分類も利用することができる。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <source>

繰り返し要件: <source>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <source>要素は、LangString として表される。LangString は 1000 キャラクターの SPM を持つ（詳細はセクション 4.2.11.2: *LangString* データタイプ参照）。

例：

```
<lom>
  <classification>
    <purpose>
      <source>LOMv1.0</source>
      <value>skill level</value>
    </purpose>
    <taxonPath>
      <source>
        <string language="en-US">ADL SCORM Concepts</string>
      </source>
    </taxonPath>
  </classification>
</lom>
```

コード例 4-73

4.2.10.2.2. <taxon>要素

<taxon>要素は、分類内のある特定の用語を表わす。分類用語は、定義されたラベルもしくは用語を持つノードである。分類用語は、標準化されたリファレンスに対して英数字の指定もしくは識別子をもつことがある。ラベルとエントリのどちらかもしくは両方が、特定の分類用語を指定するのに利用される。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <taxon>

繰り返し要件: <taxon>要素は、0 ないしそれ以上現れなくてはならない。<taxon>要素の SPM は 15 である。

データタイプ: <taxon>要素は、親要素である。親要素は自身に関連付けられた値を持たない。親要素は、他の要素に対してコンテナとしての役割を果たす。<taxon>要素は、以下の子要素を含む：

- <id>
- <entry>

例：

```
<lom>
  <classification>
    <purpose>
      <source>LOMv1.0</source>
      <value>skill level</value>
    </purpose>
    <taxonPath>
      <source>
        <string language="en-US">ADL SCORM Concepts</string>
      </source>
      <taxon>
        <id>I</id>
        <entry>
          <string language="en-US">Content Aggregation Model</string>
        </entry>
      </taxon>
    </taxonPath>
  </classification>
</lom>
```

コード例 4-74

4.2.10.2.2.1. <id>要素

<id>要素は、分類用語の識別子を記述する。例えば、Id は分類元によって提供される文字もしくは数字の組み合わせであることがある。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <id>

繰り返し要件: <id>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <id>要素の値は、文字列要素として表現される。文字列は 100 キャラクターの SPM を持つ（詳細はセクション 4.2.11.1: 文字列データタイプ参照）。

例：

```
<lom>
  <classification>
    <taxonPath>
      <source>
        <string language="en-US">ADL SCORM Concepts</string>
      </source>
      <taxon>
        <id>I</id>
        <entry>
          <string language="en-US">Content Aggregation Model</string>
        </entry>
      </taxon>
    </taxonPath>
  </classification>
</lom>
```

コード例 4-75

4.2.10.2.2.2. <entry>要素

<entry>要素は、分類用語のテキストラベルを含む。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <entry>

繰り返し要件: <entry>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <entry>要素の値は、文字列要素として表現される。文字列は、500 キャラクターの SPM を持つ (詳細はセクション 4.2.11.1: *文字列データタイプ*参照)。

例:

```
<lom>
  <classification>
    <purpose>
      <source>LOMv1.0</source>
      <value>skill level</value>
    </purpose>
    <taxonPath>
      <source>
        <string language="en-US">ADL SCORM Concepts</string>
      </source>
      <taxon>
        <id>I</id>
        <entry>
          <string language="en-US">Content Aggregation Model</string>
        </entry>
      </taxon>
    </taxonPath>
  </classification>
</lom>
```

コード例 4-76

4.2.10.3. <description>要素

<description>要素は、SCORM コンテンツモデル部品における学問分野、アイディア、スキルレベル、教育目標など特定の分類の目的の説明を含む。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <description>

繰り返し要件: <description>要素は、0 ないし 1 回現れなくてはならない。

データタイプ: <description>要素は、LangString として表される。LangString は、2000 キャラクターの SPM を持つ (詳細はセクション 4.2.11.2: *LangString データタイプ*参照)。

例：

```
<lom>
  <classification>
    <description>
      <string language="en-US">Describing and packaging SCOs in a SCORM
Content Package</string>
    </description>
  </classification>
</lom>
```

コード例 4-77

4.2.10.4. <keyword>

<keyword>要素は、SCORM コンテンツモデル部品における学問分野、アイディア、スキルレベル、教育目標など特定の分類のキーワードおよびフレーズを含む。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現: <keyword>

繰り返し要件: <keyword>要素は、0 ないしそれ以上現れなくてはならない。<keyword>要素の SPM は 40 である。

データタイプ: <keyword>要素は LangString として表される。LangString は、1000 キャラクタの SPM を持つ (詳細はセクション 4.2.11.2: *LangString* データタイプ参照)。

例：

```
<lom>
  <classification>
    <keyword>
      <string language="en-US">Packaging SCOs</string>
    </keyword>
  </classification>
</lom>
```

コード例 4-78

4.2.11. 共通データタイプ

IEEE LOM は、いくつかの共通データタイプを含む。これらのデータタイプは、個別 LOM 要素によって保持された値の構成を記述するのに利用される。以下のセクションは LOM データタイプおよびその特性を定義する。

4.2.11.1. 文字列データタイプ

文字列は、言語で解釈されないキャラクターのセットを把握するために利用されるデータタイプである。このデータタイプで提示されるキャラクターは、ISO/IEC 10646-1:2000 [10]でサポートされるキャラクターである。ISO 10646 規格は、電子情報の交信および交換に対して、統一されたキャラクターコーディング規格を提供する。

4.2.11.2. LangString データタイプ

LangString は、一つ以上の文字列を提示するデータタイプである。言語は文字列で識別され、文字列はその言語で表されている。LangString 値、は翻訳、代替記述など複数の意味的に同等な文字列を含むことがある。

XML 名前空間: <http://ltsc.ieee.org/xsd/LOM>

XML 名前空間接頭辞: lom

XML バインディング表現:

- `<string language="language-code">Textual characterstring represented in the defined language</string>`

`<string>`要素は、人言語の実際のフレーズを含む。`<string>`要素値の SPM 値は親要素によって決定される(セクション 1.2: *IEEE 1484.12.1-2002 Learning Object Metadata*で定義された LangString 要素参照)。

属性:

- `language` - `<string>`要素の内容の人言語を表す。言語属性は、100 キャラクターの SPM を持った文字列として表される。言語属性はオプションである。`<metaMetadata>`要素の `<language>`要素は、全ての LangStrings 値のデフォルト言語を表す。個別 `<string>`要素の中で、言語属性がない場合、列によって保持された値は、`<metaMetadata>`要素の `<language>`要素によって定義された言語で表される。言語属性は、それなしでは情報が失われるので、重要である。文字列の値空間は、ISO 10646-1 から来る。この規格は、各独立キャラクターに対してキャラクターコードから成り立つだけである。この規格は、異なった言語から来るキャラクターに対するコードから成り立つ。いくつかの言語が、同一キャラクターコードで表される同一のキャラクターを共有するので、言語情報は重要である。

繰り返し: `<string>`要素は、(10 の SPM `<string>`要素を持った)親要素内で 0 回以上発生する

例：

```
<general>
  <keyword>
    <string language="en">metadata</string>
    <string language="fr"> métadonnées</string>
  </keyword>
</general>
```

コード例 4-79

4.2.11.3. 語彙データタイプ

語彙データタイプを持つ特定の要素がある。語彙は、適切な値の推奨された（いくつかのケースでは要求される）一覧である。語彙データタイプは、ソース/値のペアとして表される。これは、各語彙に対してソース（もしくは所有者）および値（実際の語彙トークン）があることを意味する。LOM に定義されたこれらの語彙に対して、ソースは “LOMv1.0” と要求される。

データタイプ： 語彙データタイプは、二つの要素からなる統合データタイプである：

- **<source>**： 語彙値のソースもしくは所有者を示す。LOM 語彙の利用を要求する要素に対して、<source>要素は “LOMv1.0” の値を持つ。LOM 語彙の利用が必須でない要素に対して、<source> 値は実装で定義されたどんな文字列に設定されても構わない。文字列は、1000 キャラクターの SPM を持つ。
- **<value>**： ソースに定義される実際の値。<source>が LOMv1.0 であれば、値は LOM で定義されたリストから来る。LOM 語彙の利用が必須でない要素に対して<value>は<source>に定義される。 <value>要素の値は SPM の 1000 キャラクターを持つ。

繰り返し： <source>および<value>要素は、語彙データタイプである親要素内で 1 回だけ発生する。

例：

```
<lom>
<!--SCORM Restricted Vocabulary example-->
  <rights>
    <cost>
      <source>LOMv1.0</source>
      <value>yes</value>
    </cost>
  </rights>
<!--SCORM Best Practice Vocabulary example -->
  <educational>
    <learningResourceType>
      <source>ADL</source>
      <value>simple questionnaire</value>
    </learningResourceType>
  </educational>
</lom>
```

コード例 4-80

4.2.11.4. DateTime データタイム

DateTime データタイプは、時間のポイントを最低 1 秒の精度で記述するのに利用される。

データタイプ： DateTime データタイプは二つの要素からなる統合データタイプである：

- <dateTime>: 時間のポイントの文字列表示. 文字列は, 200 キャラクターの SPM を持つ.
- <description>: 時間のポイントの記述表示. <description>要素は, LangString データタイプである(詳細はセクション 4.2.11.2: *LangString* データタイプ参照). LangString は, 1000 キャラクターの SPM を持つ.

フォーマット： <dateTime>要素のフォーマットは, ISO8601:2000 に従って表される

YYYY[-MM][-DD[Thh[:mm[:ss[.s[TZD]]]]]]]

各々は下記の通り：

- YYYY = 4 桁の年 (>=0001)
- MM = 2 桁の月 month (01 through 12)
- DD = 2 桁の日 (01 through 31)
- hh = 2 桁の時間 (00 through 23)
- mm = 2 桁の分 (00 through 59)
- ss = 2 桁の秒 (00 through 59)
- s = 1 以上の桁数, 小数点以下の秒を表す
- TZD = タイムゾーン指定子
- 最低 4 桁の年はなければならない. DateTime の追加の部分が含まれている場合, 文字キャラクター “-”, “T”, “:” および “.” は<dateTime>に対するキャラクター語彙表示の 1 部である

繰り返し： <dateTime>および<description>要素は親要素内で 0 回か 1 回発生する.

例：

```
<lom>
  <lifeCycle>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>author</value>
      </role>
      <entity>BEGIN:VCARD\nFN:Joe AuthorEND:VCARD</entity>
      <entity>BEGIN:VCARD\nFN:Mary AuthorEND:VCARD</entity>
      <date>
        <dateTime>2002-12-12</dateTime>
        <description>
          <string language="en">A description for the date</string>
        </description>
      </date>
    </contribute>
  </lifeCycle>
</lom>
```

コード例 4-81

4.2.11.5. Duration データタイプ

Duration データタイプは時間の期間を最低 1 秒の精度で記述するのに利用される.

データタイプ： Duration データタイプは二つのデータタイプから成る統合データタイプである：

- <duration>: 時間インターバルの文字列表示. 文字列は, 200 キャラクタの SPM-を持つ
- <description>: 時間インターバルの記述表示. <description>要素は, LangString データタイプである(詳細はセクション 4.2.11.2: *LangString* データタイプ 参照). LangString は, 1000 キャラクタの SPM を持つ

フォーマット: <duration>要素のフォーマットは ISO8601:2000 に従って表される:

P[yY][mM][dD][T[hH][nM][s[.s]S]]

各々は下記の通り:

- y = 年の数
- m = 月の数
- d = 日の数
- h = 時間の数
- n = 分の数
- s = 秒の数もしくは小数点以下の秒
- 文字キャラクター指定子 “P”, “Y”, “M”, “D”, “T”, “H”, “M” および “S” は, 対応する値がゼロでなければ, 表示されなければならない.

繰り返し: <duration>および<description>要素は親要素内で 0 回か 1 回発生する.

例:

```
<lom>
  <educational>
    <typicalLearningTime>
      <duration>PT1H30M</duration>
      <description>
        <string language="en">It takes this long to complete the
course</string>
      </description>
    </typicalLearningTime>
  </educational>
</lom>
```

コード例 4-82

4.2.11.6. VCard データタイプ

vCard タイプは, 実態 (個人もしくは組織) を記述するのに利用される. vCard は, 伝統的なビジネスカードで一般的に見うけられる個人情報の交換を自動化する. vCard 仕様[9]は, パーチャル電子ビジネスカードを定義する. vCards は名前, 住所, 電話番号, イーメールアドレスなどの情報を保存できる.

以下の要素が, 有効な vCard フォーマットにある:

- Lifecycle.Contribute.Entity
- Meta-Metadata.Contribute.Entity
- Annotation.Entity

例：

```
<annotation>
  <entity>BEGIN:VCARD\nFN:Joe Friday\nTITLE:Area
Adminstrator\,Assistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</e
ntity>
</annotation>
```

コード例4-83

4.3. LOM XML スキーマバリデーションアプローチ

LOM XML バインディングは、どのように XML でメタデータインスタンスを作成するかを記述するルール集合にすぎない。XML Schema Definition (XSD) ファイルは、これらのルールを記述し実施するのに利用される。しばしば XML Schema Definition で表現できない特定のルールがある。これらのケースでは、これらのルールは、IEEE ドラフト規格で見うけられる標準テキストで記述される。IEEE は、LOM XML バインディングに対していくつかのバリデーションアプローチを提供する。これらのバリデーションアプローチは、いくつかの代替 XML スキーマを提供するために構築された。作成された各々のアプローチは、異なった結合ルールをサポートし、実施する。全てのケースで XSD ファイルは、LOM メタデータインスタンスを承認するには充分でない。LOM メタデータインスタンスは LOM XML バインディングに全てのケースで準拠する：

LOM XML Binding = “XML Schema Definition” + “Normative Standard”

IEEE は、異なったバリデーションアプローチをサポートするスキーマを提供する。これらのアプローチは、LOM メタデータインスタンスのバリデーションに利用されることができ、利用されるアプローチに基づいて必要とされるバリデーションステップがあることがある。以下のセクションは、異なったバリデーションアプローチ、スキーマドライバーおよびこれらの SCORM との関係を記述する。

4.3.1. 厳格なスキーマバリデーションアプローチ

厳格なスキーマバリデーションアプローチの主な目的は、LOM で定義された通りに、厳格に要件を実施することである。厳格なスキーマバリデーションアプローチおよび対応する XML Schema Definition (XSD) ファイルは以下の特徴を持つ：

- ユニーク性制約をサポートする。LOM で 0 回か 1 回発生するという繰り返し要件を定義された要件に対して、厳格なスキーマバリデーションアプローチは、この制約を実施する。
- LOM 語彙のみ。厳格なスキーマバリデーションアプローチは、LOM で定義された語彙トークンを利用する LOM メタデータインスタンスだけを許す。
- 拡張なし。厳格なスキーマバリデーションアプローチは、LOM への拡張をサポートしない。

このアプローチが、LOM と LOM 語彙の利用への拡張をサポートしないという事実のため、アプローチは有効で、厳格に適合した LOM メタデータインスタンス（IEEE によって定義されている）を許す。

組織が、LOM に定義された基本的な要素を拡張しないと方針を定義し、LOM に定義された語彙の利用を実施するなら、ADL は、厳格なスキーマバリデーションアプローチの利用を推奨する。これは全てのバリデーションアプローチで、最も語彙の相互運用可能性の高いものを保証することになる。

4.3.2. カスタムスキーマバリデーションアプローチ

カスタムスキーマバリデーションアプローチの主な目的は、語彙およびデータ要素への拡張をサポートするために LOM メタデータインスタンスをカスタマイズする機能を支援することである。カスタムスキーマバリデーションアプローチ（および対応する XSD ファイル）は、以下の特徴を含む：

- ユニーク性制約をサポートする。0 回か 1 回発生するという繰り返し要件持つと LOM で定義された要件に対して、カスタムスキーマバリデーションアプローチはこの制約を実施する。
- カスタマイズされた語彙。このアプローチは、LOM で定義された語彙の利用もしくは組織の語彙の利用を考慮する。バリデーションアプローチは XSD ファイルをツールで作成するのを補助するテンプレートを提供する。XSD ファイルは LOM メタデータインスタンスのバリデーションのサポートに利用されることができる。
- LOM への拡張をサポートする。このスキーマバリデーションアプローチは、LOM に定義されたデータモデルセットを拡張する機能をサポートする。これは、組織に LOM メタデータインスタンスで（他の名前空間から）要素の異なったセットを取り込むことを可能にする。

このアプローチは LOM への拡張をサポートするという事実のため、このアプローチは有効な適合した LOM メタデータインスタンス（IEEE によって定義されているように）を許す。

組織が、異なった語彙やメタデータに含まれる要素を要求する方針や慣習を含むと、ADL は、カスタムスキーマバリデーションアプローチの利用を推奨する。しかし、これは異なった組織間で語彙の相互運用可能性を許さないことを忘れないように気をつけること。高いレベルの語彙の相互運用可能性を保つために、ADL は、縦に一貫したコミュニティ（ヘルスケア産業など）がカスタムバリデーションアプローチに提供することができる交互運用可能な語彙を作ることに関して共通見解に到達するように働きかけることを推奨する。

LOM の語彙タイプの要素に対して、SCORM は、特定の要素がその語彙を利用することを要求するだけである。いくつかの要素に対して、SCORM は、定義された LOM 語彙を利用することを推奨されるベストプラクティスとするだけである。組織が、SCORM ベストプラクティス語彙要素に対して、異なった語彙を定義する必要があるれば、ADL は、カスタムスキーマバリデーションアプローチおよび XSD ファイル作成に関するガイダンスの利用を推奨する。これは、ツールがメタデータインスタンスを正しく認証することを認めている。

4.3.3. 柔軟なスキーマバリデーションアプローチ

柔軟なスキーマバリデーションアプローチの主な目的は、他のスキーマバリデーションアプローチで定義された制約のいくつかを緩和することである。厳格なもしくはカスタムバリデーションアプローチを利用するとき、ツールを、LOM のユニーク性制約のバリデーションで助けるために人工的な属性が導入される。組織が、この人工的な要素の導入を避けたければ、柔軟なバリデーションアプローチが利用されることができる。柔軟なスキーマバリデーションアプローチ（そして対応する柔軟なスキーマドライバー）は、ユニーク性制約をチェックしないということと適合

しない LOM メタデータインスタンスを認めるということを忘れてはいけない。ツールが、全部の LOM XML バインディングルールを実施するのに、追加のステップが取られなければならない。

柔軟なスキーマバリデーションアプローチ（および対応する XML スキーマ定義ファイル）は以下の特徴を含む：

- ユニーク性制約をサポートしない。柔軟なスキーマバリデーションアプローチは、ユニーク性制約チェックを、人工的な属性の導入を除去することにより、緩和する。このために、適合しない LOM メタデータインスタンスが柔軟なスキーマバリデーションアプローチに対して認証するケースがある。LOM メタデータインスタンスの作成者は、柔軟なスキーマバリデーションアプローチを利用するときには、インスタンスは、ドラフト IEEE LOM XML Binding に従って有効であることを保証することが推奨される。
- 語彙のバリデーションなし。柔軟なスキーマバリデーションアプローチは、語彙のソースおよび値ペア関係性の制約に対するスキーマの縛りを緩和する。柔軟なスキーマバリデーションアプローチは、スキーマバリデーションプロセスを単純化する。しかし、抑止力がないため、適合された LOM メタデータインスタンスを保証しない。アプリケーションは、他の手段で語彙のソースおよび値ペア関係の制約を有効にしなければならない。
- LOM への拡張をサポートする。このバリデーションアプローチは、LOM に定義されたデータモデルセットを拡張する機能をサポートする。これは、組織に LOM メタデータインスタンスで（他の名前空間から）要素の異なったセットを取り込むことを可能にする。

柔軟なスキーマバリデーションアプローチは、LOM メタデータインスタンスが IEEE 要件に適合することを立証するのに、バリデーションツールの外で、より多くの処理を要求する。ADL は、柔軟なスキーマバリデーションアプローチの代わりに他のスキーマバリデーションアプローチの一つが利用されることを推奨する。

4.4. メタデータ拡張

いくつかのケースでは、組織は、LOM で定義されたメタデータ要素のコアセットが、SCORM コンテンツモデル部品を記述するのに充分でないと認識することがある。組織は、これらの部品を記述するのに利用されるように要求されるメタデータ拡張を持つことがある。LOM 内で認められる拡張メカニズムは、現在 2 種類ある：

- XML 要素拡張。最初のメカニズムは、LOM データモデル要素の拡張を考慮している。メタデータインスタンスに、追加要素を追加することが認められている。例えば、組織が、SCORM コンテンツモデルの知的所有権や条件あるいは利用に関する追加情報を持てば、組織は、Rights カテゴリに要素を追加することができる。現在、デジタル著作権の管理に関するいくつかの研究および開発が進んでいる。より堅固で豊富な権利のセットを表わす要素のセットが開発される可能性がある。これが、理想では、Rights カテゴリを拡張して SCORM コンテンツモデル部品を記述するのに利用されることが可能となる。
- 語彙拡張。語彙リストを持ついくつかの IEEE 要素に対して、SCORM は、その語彙の利用を推奨するが、これは単に推奨であり、メタデータインスタンスが、その語彙を利用するように要求されてはいない。組織が、SCORM ベストプラクティス語彙としてリストされた要素の代わりに、異なった語彙を利用する必要性があれば、組織はいくつかの代替手段を持つ。組織が、語彙のバリデーションを実施したければ、ADL は、カスタムスキーマバリデーションアプローチの利用を推奨する。（詳細はセクション 4.4.2: *Vocabulary Extension* 参照）。厳格なスキーマバリデーションアプローチは、厳格な LOM 語彙ペアしか認証しないので、利用できない。

拡張を利用する際のいくつかの留意点：

1. 拡張要素を作成するとき、現在定義された IEEE LOM 要素と同一の意味を含む要素を定義することは認められていない
2. 推奨される値に依存するメタデータは、最も高いレベルの意味上の相互運用可能性を持つ。高いレベルの意味上の相互運用可能性を保つために、ADL は、縦に一貫したコミュニティ（ヘルスケア産業等々）のニーズを満たすために、拡張が必要とされるなら、縦のコミュニティは、相互運用可能な拡張を構築することに共通見解を得られるように働きかけるべきであることを推奨する

LOM は、IEEE 規格への適合の二つのタイプを区別する。LOM メタデータインスタンスが、何も拡張を含まなければ、IEEE 規格は、これを厳格に LOM メタデータインスタンスに適合するとして言及する。LOM メタデータインスタンスが、拡張されたデータ要素を含めば、IEEE 規格は、これを LOM メタデータインスタンスに適合するとしてと言及する。SCORM は、両方のタイプの適合をサポートし、LOM メタデータインスタンスが厳格に適合することを推奨する(前述の注意に基づいて)。

4.4.1. データ要素拡張

LOM がその要素でサポートしない方法で、SCORM コンテンツモデル部品を記述する方針や慣習を、組織が持つことがある。例えば、組織が、自分の学習コンテンツを記述するのに使用する堅固なデジタル著作権管理スキームをもつことがある。LOM は、そのベーススキームが拡張されることを認める。前述の通り、これはメタデータおよび学習コンテンツの意味上の相互運用可能性を低下させる可能性がある。

組織が、既存の LOM に独自の拡張を施したいときには、以下のルールが遵守されるべきである：

- LOM ベーススキームへの拡張は、LOM ベーススキームから値空間およびデータ要素のデータタイプを引き継ぐ
- 拡張は、LOM ベーススキームの統合データ要素のデータタイプもしくは値空間を定義しない
- 拡張されたデータ要素は、LOM ベーススキームのデータ要素を交換しない

要素拡張は、XML 仕様に基づいて様々な方法で処置される。ADL は、XML 拡張が組織に必要とされれば、組織が XML メタデータインスタンスバリデーションに対して利用可能な XSD ファイルを提供するように推奨する。

4.4.2. 語彙拡張

IEEE は、LOM に定義された語彙トークンセットの利用を推奨する、SCORM はこの推奨を認めるが、必要があれば、ベストプラクティス語彙と示された要素に対する組織独自の語彙トークンセットの作成および利用を認める。前述の通り、これは、メタデータおよび学習コンテンツの意味上の相互運用可能性を低下させる危険性があり、注意して利用されるべきである。高いレベルの意味上の相互運用可能性を保つために、ADL は、縦に一貫しているコミュニティ（ヘルスケア産業等々）が、相互運用可能な語彙のセットを構築することに共通見解をえられるようにすることを推奨する。縦に一貫した業界で、新しい語彙トークンを作成する必要があるれば、その業界内の他社と共同で合意された語彙トークンセットを作成することが適切である。

ADL は、拡張語彙を表すのにカスタムスキームバリデーションアプローチの利用を推奨する。組織定義の語彙トークンを作成し利用するために、IEEE カスタムスキームバリデーションアプローチは、LOM XML Binding で定義されたものと同様の XML バインディングを要求する。

4.5. メタデータと SCORM コンテンツモデル要素

これまで、このドキュメントのメタデータセクションは、LOM 情報モデルについて、どのようにこの情報モデルが XML にバインドされるか、および、組織の方針やビジネスニーズを満たすために LOM を拡張する方法の詳細を説明してきた。このセクションでは、SCORM コンテンツモデル要素（コンテンツアグリゲーション、コンテンツオーガニゼーション、アクティビティ、SCO およびアセット）にメタデータを関連付けるために組織が選択できる方法のひとつを説明する。

SCORM は、メタデータの SCORM コンテンツモデル要素への対応付けを、SCORM コンテンツパッケージマニフェストによって可能としている。それぞれの SCORM コンテンツモデル要素はマニフェスト内で表現される。マニフェストはメタデータ（3.4 コンテンツパッケージの作成参照）をこれらの要素に対応付ける方法を提供している。

- **コンテンツアグリゲーションメタデータ：** コンテンツアグリゲーションメタデータはコンテンツアグリゲーション（すなわちコンテンツパッケージ）全体を記述する。コンテンツアグリゲーションメタデータを適用する目的は、コンテンツアグリゲーションの検索可能性を向上し、コンテンツアグリゲーション全体の記述的情報を提供することにある。
- **コンテンツオーガニゼーションメタデータ：** コンテンツオーガニゼーションメタデータはコンテンツオーガニゼーションを記述する。コンテンツオーガニゼーションメタデータを適用する目的は、例えば、コンテンツリポジトリ中での検索可能性を向上し、コンテンツオーガニゼーションで定義されるコンテンツ構造全体の記述的情報を提供することにある。
- **アクティビティメタデータ：** アクティビティメタデータは個々のアクティビティを記述する。アクティビティメタデータを適用する目的は、コンテンツリポジトリ中でアクティビティをアクセス可能（検索可能）とすることにある。メタデータはアクティビティ全体を記述しなくてはならない。
- **SCO メタデータ：** メタデータは、使用形態に依らない SCO 中のコンテンツの記述的情報を提供するために SCO に適用することができる。このメタデータはコンテンツの再利用と検索可能性を向上させるために用いる。
- **アセットメタデータ：** メタデータは、教材コンテンツ中の使用形態に依らないアセットの記述的情報を提供するためにアセットに適用することができる。このメタデータは、例えば、コンテンツ作成においてリポジトリ内でのコンテンツの再利用と検索可能性を向上させるために用いる。

先に述べたように、SCORM では SCORM コンテンツパッケージでメタデータを定義することは要求されない。組織や実践コミュニティは、彼らのユースケースと要件を特定して、メタデータを付与するためにそのようなメカニズムを使用することが推奨される。システムは、コンテンツパッケージで定義されたメタデータをサポートしてもしなくてもよい。例えば、ある LMS はコンテンツパッケージに付与されたメタデータを検索し、内部のコンテンツカタログを満たすためにメタデータを用いるように作られていてよい。重要なことは、このような動作が SCORM で要求されているのではない、ということである。コンテンツ開発者にとって重要なのは、メタデータをコンテンツパッケージに付与する目的と、それを最終的にシステムがどのように使用するかである。このようなユースケースと要求条件は SCORM では規定されていない。

4.5.1. コンテンツアグリゲーション記述メタデータ

コンテンツアグリゲーションレベルのメタデータは、パッケージ（コンテンツアグリゲーション）全体を記述するのに利用される。もし、メタデータがコンテンツアグリゲーションを記述するために用意されている場合、メタデータは先に概要を示した要件に適合しなくてはならない。

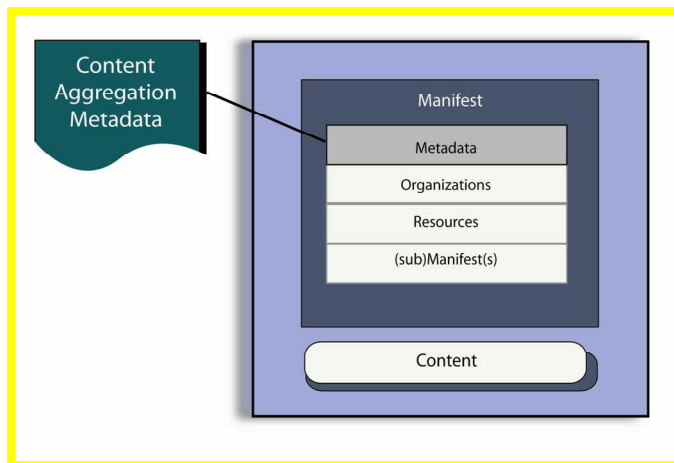


図4.5.1.1a: コンテンツアグリゲーションメタデータの適用

以下の例は、コンテンツパッケージのマニフェストでメタデータインラインが含まれることを示す。これは、マニフェストにメタデータを適用する一つの方法である。このメカニズムは、残りのメタデータアプリケーションプロファイルでも利用可能である。

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>2004 3rd Edition</schemaversion>
    <lom xmlns="http://ltsc.ieee.org/xsd/LOM">
      <general />
      <classification />
      <annotation />
      <lifeCycle />
      <technical />
      <metaMetadata />
      <educational />
      <relation />
      <rights />
    </lom>
  </metadata>
</manifest>
```

コード例4-85 インラインネームスペースメタデータ

以下の例は、メタデータを記述するための<adlcp:location>要素の使用例を示す。<adlcp:location>要素は、パッケージのルートに対して相対的なメタデータのロケーションを記述する。


```

<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>2004 3rd Edition</schemaversion>
    <adlcp:location>contentAggregationMetadata.xml</adlcp:location>  </metadata>
  </manifest>

```

コード例4-86 参照メタデータへの<adlcp:location>の使用

どちらの例も、コンテンツアグリゲーションメタデータアプリケーションプロファイルに対してどのようにXMLが作成され、マニフェストファイルのどこに位置されるかを示す。

4.5.2. コンテンツオーガニゼーション記述メタデータ

コンテンツオーガニゼーションレベルのメタデータは、コンテンツオーガニゼーションを記述する。このメタデータは、コンテンツレポジトリもしくは同様のシステム内での再利用および発見可能性を向上するのに利用される。これは、コンテンツオーガニゼーションに関する記述情報を提供することにより達成される。コンテンツオーガニゼーションメタデータは、コンテンツオーガニゼーション全体の情報である。コンテンツオーガニゼーションは何のためにあり、誰が利用し、誰がコントロールするのか、およびコンテンツオーガニゼーションのタイトル、名称、説明、バージョンなど外部から検索できる情報を記述する。

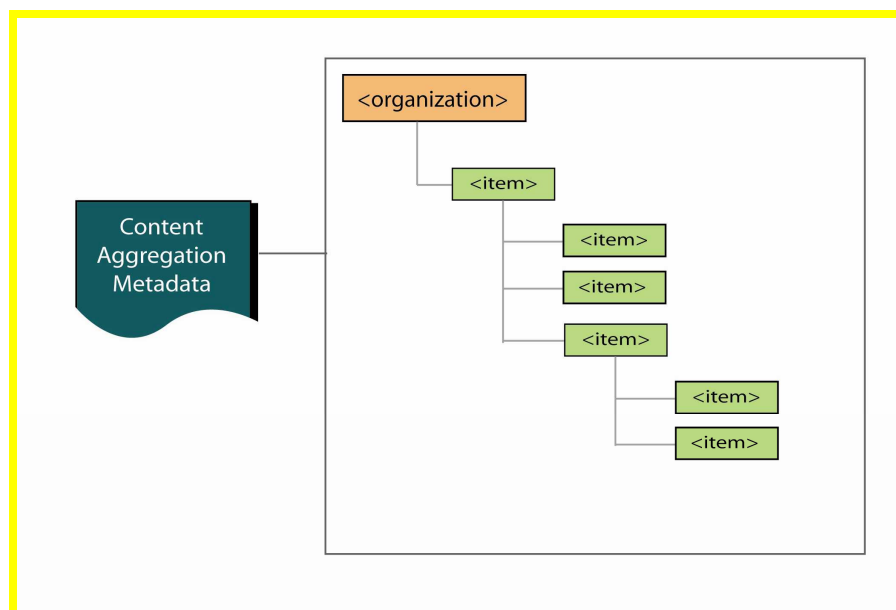


図4.5.1.2a: コンテンツオーガニゼーションメタデータの適用

コンテンツオーガニゼーションメタデータは、インラインメタデータとして適用するか、もしくは<adlcp:location>要素で参照することができる。具体的には、パッケージのマニフェストで、<organization>要素に適用される。単純化する目的で、以下の例は、記述されているメタデータへの参照を掲示するために、<adlcp:location>要素を利用する。これは簡素化のために行われることに留意しなければならない。インラインメタデータは、<adlcp:location>要素が見つかる全ての個所で、代わりに用いることができる。

```

<organizations>
  <organization>
    <title>Introduction to the SCORM</title>
    <item>...</item>
    <item>...</item>
    <metadata>
      <adlcp:location>contentAggregationMetadata.xml</adlcp:location>
    </metadata>
  </organization>
</organizations>

```

コード例 4-87 オーガニゼーションメタデータの例

4.5.3. アクティビティ記述メタデータ

アクティビティレベルメタデータは、アクティビティ(imsmanifest.xml ファイルで<item>要素として提示される)を記述するメタデータである。このメタデータは、コンテンツレポジトリもしくは同様のシステム内で再利用および発見可能性を向上し、アクティビティに関する記述情報を提供するのに利用される。アクティビティメタデータは、一般的には、文脈に依存する形で、アクティビティ全体を何のために誰が利用し誰がコントロールするかといった情報を含む。

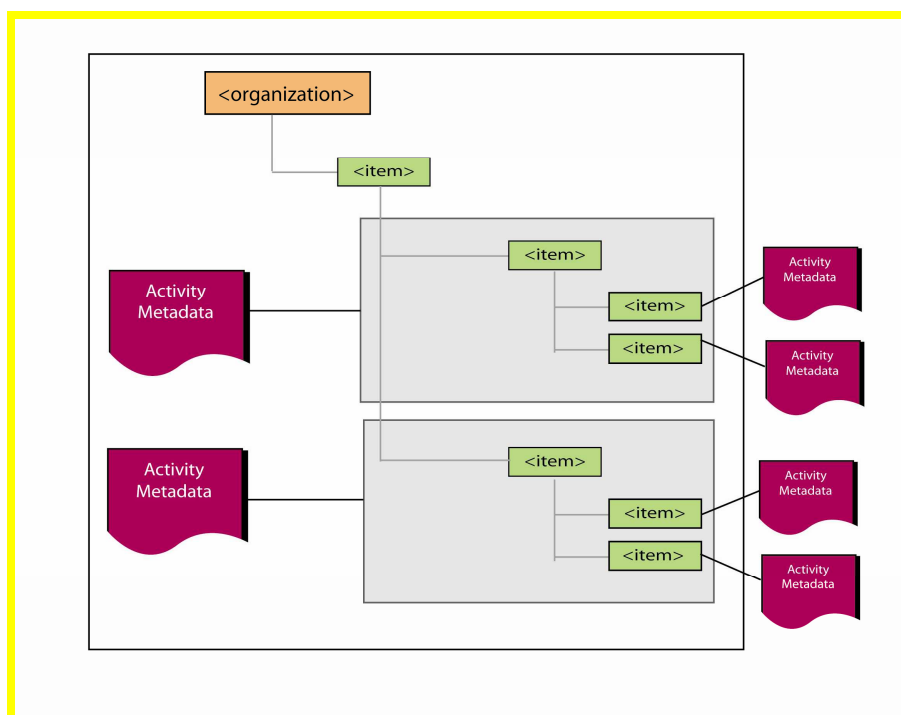


図 4.5.1.3a: アクティビティメタデータの適用

アクティビティメタデータは、コンテンツパッケージのマニフェストの<item>要素に適用される。

```

<organizations>
  <organization>
    <title>Introduction to the SCORM</title>
    <item>
      <title>SCORM 101</title>
      <metadata>
        <adlcp:location>activityMetadata.xml</adlcp:location>
      </metadata>
    </item>
  </organization>
</organizations>

```

コード例 4-88 アクティビティメタデータの例

4.5.4. SCO 記述メタデータ

メタデータは SCO に関連付けることが可能であり、学習リソースに関する特定の文脈に依存しない記述情報を提供する。このメタデータは、そのような学習リソースの再利用および発見可能性を向上させる。SCO メタデータは、SCO を記述する、特定のコンテンツオーガニゼーション構成に関連しない（すなわち文脈に依存しない）メタデータである。このメタデータは、コンテンツのタイトル、記述、作成日、バージョン日付などを外部から検索することが可能な情報を含む。

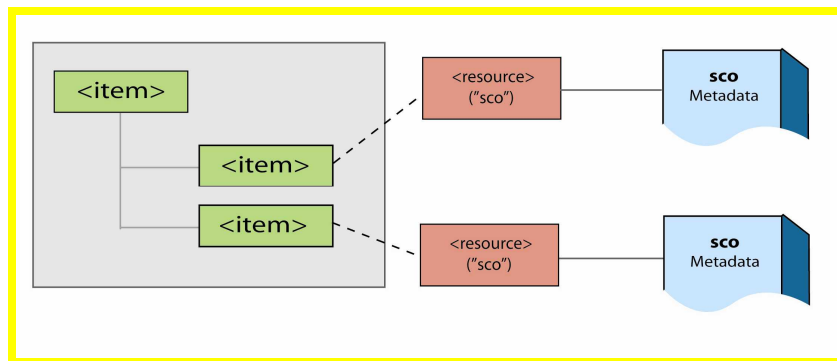


図 4.5.1.4a: SCO メタデータの適用

SCO メタデータは、コンテンツパッケージのマニフェストで `adlcp:scormType="sco"` 属性をもつ `<resource>` 要素（すなわち SCO リソース）に適用される。

```

<resources>
  <resource type="webcontent" adlcp:scormType="sco" href="sco1.htm">
    <metadata>
      <adlcp:location>SCOMetadata.xml</adlcp:location>
    </metadata>
  </resource>
</resources>

```

コード例 4-89 リソースメタデータの例

4.5.5. アセット記述メタデータ

メタデータは、イラストレーション、ドキュメントもしくは媒体ストリームといったアセットに関連付けることが可能である。アセットメタデータは、SCORM アセットに関する学習コンテンツ

から独立した記述情報を提供する。このメタデータは、学習コンテンツ作成時に、主にそのようなアセットの再利用および発見可能性を向上するために利用される。アセットメタデータは、外部から文脈に依存しない形で、タイトル、作成およびバージョン日付により検索でき、さらに、共有可能なアセットの検索可能なレポジトリを作成するのに用いることが可能なメタデータである。

アセットメタデータは、コンテンツパッケージのマニフェストで `adlcp:scormType="asset"` 属性を持った `<resource>` 要素（すなわちアセットリソース）に適用される。アセットメタデータは、`<resource>` 要素の子要素の `<file>` 要素に適用することができる。

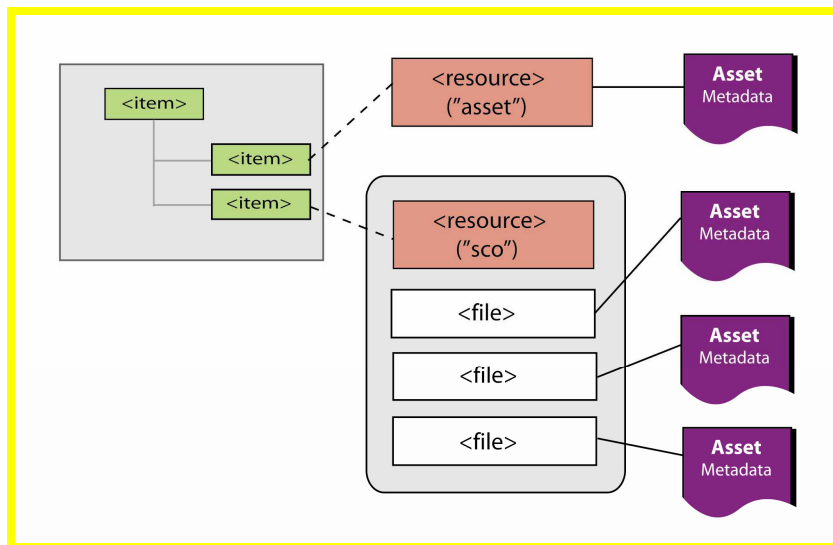


図4.5.1.5a: アセットメタデータの適用

以下の例はアセットを表すリソースのアセットメタデータを示す。

```
<resources>
  <resource type="webcontent" adlcp:scormType="asset" href="asset.htm">
    <metadata>
      <adlcp:location>assetMetadata.xml</adlcp:location>
    </metadata>
  </resource>
</resources>
```

コード例4-90 リソースアセットメタデータの例

以下の例は、`<file>` 要素に対するアセットメタデータを示す

```
<resources>
  <resource type="webcontent" adlcp:scormType="asset" href="asset.htm">
    <file href="asset.htm">
      <metadata>
        <adlcp:location>assetMetadata.xml</adlcp:location>
      </metadata>
    </file>
  </resource>
</resources>
```

コード例4-91 ファイルアセットメタデータの例

セクション 5

SCORM シーケンシングおよび プレゼンテーション

このページは空白である

5.1. シーケンシングおよびプレゼンテーション

このセクションは、特定のシーケンシング戦略を XML でどのように表現するかを記述する。この XML は、アクティビティに対するシーケンシング情報を定義するために、IMS マニフェストファイルに置かれる。シーケンシング情報を作成するのには、主に二つの方法がある：

- **<sequencing>要素**. **<sequencing>要素**は、あるアクティビティに対して全ての必要な情報を含む。
- **<sequencingCollection>要素**. **<sequencingCollection>要素**は、いくつかのアクティビティで再利用されるシーケンシング情報の集合を集約するのに利用される

アクティビティは、マニフェスト内で<item>要素もしくは<organization>要素として表現される。<sequencing>要素は、<item>もしくは<organization>要素の子になるが、例外が一つある。リーフ<item>要素が、(サブ)マニフェストを参照すると、そのリーフ<item>要素は、自身に関連付けられたシーケンシング情報を持つことができない。

シーケンシング情報および戦略に関するより詳細については、SCORM SN ブックに記述されている。

5.1.1. <sequencing>要素

シーケンシング情報は、単一の <sequencing>要素を階層のアイテムに関連付けることにより、階層構造でアイテムと関連付けられる。IMS コンテンツパッケージの文脈で、この関連付けは、<item>要素もしくは<organization>要素内に<sequencing>要素を含めることで行なわれる。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <sequencing>

データタイプ: <sequencing>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<sequencing>要素は以下の要素/属性を含む：

属性:

- **ID (オプション)** - このシーケンシング情報の集合に割り当てられた固有の識別子。ID 属性は、<sequencingCollection>要素の子である<sequencing>要素でだけ許されている。ID 属性は、<item>もしくは<organization>要素の子である<sequencing>要素では許されていない。
XML Data Type: xs:ID.
- **IDRef (オプション)** - シーケンシング情報に割り当てられた固有の識別子への参照 (<sequencing>要素の ID 属性)。IDRef は、同一の XML ドキュメントの中で定義された再利用可能なシーケンシング情報にリンクするのに用いられる。XML Data Type: xs:IDREF.

要素:

-
- <controlMode>
 - <sequencingRules>
 - <limitConditions>
 - <auxiliaryResources>
 - <rollupRules>
 - <objectives>
 - <randomizationControls>
 - <deliveryControls>
 - <adlseq:constrainedChoiceConsiderations>
 - <adlseq:rollupConsiderations>

繰り返し: <sequencingCollection>要素が存在すれば, <sequencingCollection>要素内で1回以上存在する。IMS コンテンツパッケージ内の各<item>もしくは<organization>では0回か1回存在する。

例:

```
<item identifier="INTRO" identifierref="RESOURCE_INTRO">
  <title>Photoshop Introduction</title>
  <imsss:sequencing>
    <imsss:limitConditions attemptLimit="1"/>
    <imsss:rollupRules rollupObjectiveSatisfied="false"/>
  </imsss:sequencing>
</item>
```

コード例5-1

5.1.2. <controlMode>要素

<controlMode>要素は、アクティビティに対して指定されたシーケンシング動作のタイプの記述を含むシーケンシングコントロールモード情報のコンテナである[5]。この要素は、許されたシーケンシング要求のタイプを取り扱う情報を取り込む。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <controlMode>

データタイプ: <controlMode>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<controlMode>要素は以下の要素/属性を含む：

属性:

- choice (オプション。デフォルト値 = true) - このアクティビティの子を対象とする choice シーケンシング要求が許される（もしくは、値が false なら許されない）ことを示す[5]。XML Data Type: xs:boolean.
- choiceExit (オプション。デフォルト値 = true) - choice シーケンシング要求が処理されて、このアクティビティのアクティブな子が終了するの許される（もしくは、値が false なら許されない）ことを示す[5]。XML Data Type: xs:boolean.
- flow (オプション。デフォルト値= false) - このアクティビティの子を対象とする flow シーケンシング要求（訳注：continue および previous 要求）が許される（もしくは、値が false なら許されない）ことを示す[5]。XML Data Type: xs:boolean.
- forwardOnly (オプション。デフォルト値 = false) - このアクティビティの子に対して（アクティビティツリー探索で）後方のアクティビティをターゲットとすることは許されない（もしくは、値が false なら許される）ことを示す[5]。XML Data Type: xs:boolean.
- useCurrentAttemptObjectiveInfo (オプション。デフォルト値 = true) - アクティビティの子の学習目標進捗情報は、アクティビティへの現在の試行中に学習目標進捗情報が記録された場合だけ、ルール評価およびロールアップにおいて使用される（もしくは、値が false なら使用されない）ことを示す[5]（訳注：値が false なら現在の試行以前に記録された値でも使用する）。XML Data Type: xs:boolean.
- useCurrentAttemptProgressInfo (オプション。デフォルト値= true) - アクティビティの子のアテンプト進捗情報は、アクティビティへの現在の試行中にアテンプト進捗情報が記録された場合だけ、ルール評価およびロールアップにおいて使用される（もしくは、値が= false なら使用されない）ことを示す[5]（訳注：値が false なら現在の試行以前に記録された値でも使用する）。XML Data Type: xs:boolean.

要素:

- なし

繰り返し: <sequencing>要素内で 0 回か 1 回存在する。

例：

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:controlMode choice="false" choiceExit="false"
      flow="true" forwardOnly = "true"/>
  </imsss:sequencing>
</item>
```

コード例5-2

5.1.3. <sequencingRules> 要素

<sequencingRules>要素は、シーケンシングルール記述に対するコンテナである。各ルールは、アクティビティに対するシーケンシング動作を表わす。各アクティビティのシーケンシングルールは無制限であり、どの区分内でもルールは列挙された順序で評価される[5]。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <sequencingRules>

データタイプ: <sequencingRules>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<sequencingRules>要素は以下の要素/属性を含む：

属性:

- なし

要素:

- <preConditionRule>
- <exitConditionRule>
- <postConditionRule>

繰り返し: <sequencing>要素で 0 回以上存在する

例:

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "disabled"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-3

5.1.3.1. <preConditionRule>要素

<preConditionRule>要素はシーケンシングの判断と特定のアクティビティの配信を制御する動作の記述のためのコンテナである。そのような動作を含むルールはアクティビティを配信するかどうかが決定するために用いられる[5]。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <preConditionRule>

データタイプ: <preConditionRule>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<preConditionRule>要素は以下の要素/属性を含む：

属性:

- なし

要素:

- <ruleConditions>
- <ruleAction>

繰り返し: <sequencingRules>要素で 0 回以上存在する

例:

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "disabled"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-4

5.1.3.1.1. <ruleConditions>要素

<ruleConditions>要素は、プリコンディション、ポストコンディション、そして、終了条件ルール（訳注：終了アクションルール）に適用される条件のセットのコンテナである。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <ruleConditions>

データタイプ: <ruleConditions>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<ruleConditions>要素は以下の要素/属性を含む:

属性:

- conditionCombination (オプション。デフォルト値 = all)。この属性は、ルールを評価するのに、どのようにルール条件 (<ruleCondition>) が組み合わせられるかを示す[5]。
XML Data Type: xs:token.
 - o **all:** 個々のルール条件が全て true と評価される場合に限り、ルール条件は true と評価される[5]。
 - o **any:** 個々のルール条件のいずれかが true と評価されれば、ルール条件は true と評価される[5]。

要素:

- <ruleCondition>

繰り返し: <preConditionRule>, <exitConditionRule>および<postConditionRule>要素内で 0 回か 1 回存在する

例:

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "disabled"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-5

5.1.3.1.1.1. <ruleCondition>

<ruleCondition>要素は評価される条件を表す。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <ruleCondition>

データタイプ: <ruleCondition>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<ruleCondition>要素は以下の要素/属性を含む:

属性:

- **referencedObjective** (オプション): この属性はアクティビティに関連付けられたローカル学習目標の識別子を表す。この学習目標の状態が条件の評価で用いられる[5]。もし、提供されている場合、referencedObjectiveの値は、そのアクティビティに対して定義された<primaryObjective>か<objective>の objectiveID の値でなくてはならない。IMS シンプルシーケンシング仕様で定義しているとおり、referencedObjectiveの根本的なデータタイプは一意な識別子である。空文字列は参照される学習目標を一意に識別するために十分な意味的情報を持たないため referencedObjective は空文字列や、全部が (XML パーサで空文字列に変換される) 空白文字であってはならない。もし referencedObjective が省略された場合、このアクティビティの主学習目標が、正規に宣言されているか否かに依らず、ルールの評価に用いられる。
XML Data Type: xs:string.
- **measureThreshold** (オプション). 習得度に基づく条件評価において閾値として用いられる値[5]. XML Data Type: xs:decimal (範囲 -1.0000 から 1.0000 で小数点以下最低4桁の制度)
- **operator** (オプション. デフォルト値 = noOp). 条件に適用される単一論理演算子[5]. XML Data Type: xs:token.
 - **not**: ルール評価で対応する条件を否定する[5]
 - **noOp**: ルール評価で対応する条件をそのまま利用する[5]
- **condition** (必須). この属性は、ルールの実際の条件を表す[5]. XML Data Type: xs:token. 以下は、condition 属性に対して利用される語彙トークンの一覧である:
 - **satisfied**
 - **objectiveStatusKnown**
 - **objectiveMeasureKnown**
 - **objectiveMeasureGreaterThan**
 - **objectiveMeasureLessThan**
 - **completed**
 - **activityProgressKnown**
 - **attempted**
 - **attemptLimitExceeded**
 - **timeLimitExceeded**
 - **outsideAvailableTimeRange**
 - **always**

要素:

- なし

繰り返し: <ruleConditions>要素の中で1回以上存在する。

例：

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "disabled"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-6

5.1.3.1.2. <ruleAction>要素

<ruleAction>要素は、ルールが true と評価された場合に求められるシーケンシング動作である。ルールアクションは条件のタイプ (<preConditionRule>, <postConditionRule>もしくは<exitConditionRule>) により異なる。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <ruleAction>

データタイプ: <ruleAction>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<ruleAction>要素は以下の要素/属性を含む：

属性:

- action (必須。アクションが定義されていないとアクションは無視される)。アクションは、ルール条件が true と評価された場合に求められるシーケンシング動作を示す[5]。
 - action が<preConditionRule>で定義された場合、action 属性は以下のどれか一つの値を持つ：
 - skip
 - disabled
 - hiddenFromChoice
 - stopForwardTraversal
 - action が<postConditionRule>で定義された場合、action 属性は以下のどれか一つの値を持つ：
 - exitParent
 - exitAll

- **retry**
 - **retryAll**
 - **continue**
 - **previous**
- action が<exitConditionRule>で定義された場合、action 属性は以下のどれか一つの値を持つ：
 - **exit**

要素：

- なし

繰り返し： <preConditionRule>, <postConditionRule>, もしくは, <exitConditionRule>要素内で1回だけ存在する

例：

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "disabled"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-7

5.1.3.2. <postConditionRule>要素

<postConditionRule>要素は、シーケンシングの判断と特定のアクティビティの配信を制御する動作の記述のためのコンテナである。そのような動作を含むルールは、アクティビティアテンプトが終了する際に適用される [5].

XML 名前空間： <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞： imsss

XML バインディング表現： <postConditionRule>

データタイプ： <postConditionRule>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<postConditionRule>要素は以下の要素/属性を含む：

属性:

- なし

要素:

- <ruleConditions> (詳細は「5.1.3.1.1 <ruleConditions>要素」参照)
- <ruleAction> (詳細は「5.1.3.1.2 <ruleAction>要素」参照)

繰り返し: <sequencingRules>要素で 0 回以上存在する。

例:

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:postConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition="satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action="exitParent"/>
      </imsss:postConditionRule>
      <imsss:postConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "always"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "continue"/>
      </imsss:postConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-8

5.1.3.3. <exitConditionRule>要素

<exitConditionRule>要素は、シーケンシングの判断と特定のアクティビティの配信を制御する動作の記述のためのコンテナである。そのような動作を含むルールは、子孫のアクティビティアテンプトが終了する際に適用される[5]。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <exitConditionRule>

データタイプ: <exitConditionRule>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<exitConditionRule>要素は以下の要素/属性を含む:

属性:

- なし

要素:

- <ruleConditions> (詳細は「5.1.3.1.1 <ruleConditions>要素」参照)
- <ruleAction> (詳細は「5.1.3.1.2 <ruleAction>要素」参照)

繰り返し: <sequencingRules>要素で 0 回以上存在する.

例:

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <imsss:sequencing>
    <imsss:sequencingRules>
      <imsss:exitConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition="satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action="exit"/>
      </imsss:exitConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-9

5.1.4. <limitConditions>要素

現時点で、ADL は、<limitConditions>要素を注意して利用することを推奨する。時間に関する条件を評価するために、様々な課題がとり上げられている。ADL は、この要素と、そのシーケンシング、動作および戦略への影響の評価を続ける予定である。

ADL は、現在二つの制限コンディションの利用をサポートするだけである。制限コンディションは、アクティビティへの試行およびその試行で許された最大時間を取り扱う。このセクションは、試行制限属性と試行最大許容時間属性、およびその利用のための用例を記述する。

他のリミット条件についての詳細は、IMS シンプルシーケンシング仕様[5]を参照されたい。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <limitConditions>

データタイプ: <limitConditions>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<limitConditions>要素は以下の要素/属性を含む：

属性:

- attemptLimit (オプション)。この値は、アクティビティに対する試行の最大回数を示す[5]。XML Data Type: xs:nonNegativeInteger.
- attemptAbsoluteDurationLimit (オプション)。この値は、学習者がアクティビティへの単一の試行に費やせる最大時間を示す。制限は、学習者がアクティビティと相互作用している時間だけに適用し、アクティビティが中断しているときは適用しない[5]。この要素は、cmi.max_time_allowedを初期化するのに利用される (SCORM RTE ブック[2]参照)。現在、SCO が全ての時間追跡および時間制限違反に対する動作の責任は、すべて SCO にある。XML Data Type: xs:duration.

要素:

- なし

繰り返し: <sequencing>要素で 0 回か 1 回存在する。

例:

```
<item identifier="INTRO" identifierref="RESOURCE_INTRO">
  <title>Photoshop Introduction</title>
  <imsss:sequencing>
    <imsss:limitConditions attemptLimit="1"/>
  </imsss:sequencing>
</item>
```

コード例 5-10

5.1.5. <auxiliaryResources> 要素

現時点で、ADL は、<auxiliaryResources>要素を注意して利用することを推奨する。補助リソース利用の要件の定義の扱いについて様々な課題がとり上げられている（例えば、補助リソースは SCO になれるか、シーケンシング情報を補助リソースに適用するか、など）。ADL は、この要素と、そのシーケンシング、動作および戦略への影響の評価を続ける予定である。

補助リソースに関するより以上の情報については、IMS SS[5]を参照のこと。

5.1.6. <rollupRules>要素

<rollupRules>要素は、アクティビティに対して定義されたロールアップルールの集合のコンテナである。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <rollupRules>

データタイプ: <rollupRules>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<rollupRules>要素は以下の要素/属性を含む：

属性:

- **rollupObjectiveSatisfied** (オプション。デフォルト値 = true)。この属性は、アクティビティに関連付けられた学習目標の習得状態が、親アクティビティに対するロールアップに含まれること示す[5] (訳注：アクティビティのトラッキング状態情報を、親アクティビティの Rollup Actions が Satisfied および Not Satisfied のロールアップルールで参照するかどうかを示す。)。XML Data Type: xs:boolean.
- **rollupProgressCompletion** (オプション。デフォルト値 = true)。この属性は、アクティビティの試行完了状態が、親アクティビティに対するロールアップに含まれること示す[5] (訳注：アクティビティのトラッキング状態情報を、親アクティビティの Rollup Actions が Completed および Incomplete のロールアップルールで参照するかどうかを示す。)。XML Data Type: xs:boolean.
- **objectiveMeasureWeight** (オプション。デフォルト値 = 1.0000)。この属性は、親アクティビティのロールアップ時に、学習目標の正規化習得度に適用する重み付け比率を用いることを示す[5]。XML Data Type: xs:decimal (範囲：0.0000 から 1.0000。精度：小数点以下最低 4 桁)。

要素:

- <rollupRule>

繰り返し: <sequencing>要素で 0 回か 1 回存在する。

例：

```
<item identifier="MODULE2">
  <title>Module 2 -- Enhancing Images</title>
  <item identifier="PRETEST2">
    <title>Module 2 -- Pretest</title>
    <item identifier="PRETEST_QUESTION4" isvisible = "false"
      identifierref="RESOURCE_QUESTION4">
      <title>Question 4</title>
    </item>
    <item identifier="PRETEST_QUESTION5" isvisible = "false"
      identifierref="RESOURCE_QUESTION5">
      <title>Question 5</title>
    </item>
    <item identifier="PRETEST_QUESTION6" isvisible = "false"
      identifierref="RESOURCE_QUESTION6">
      <title>Question 6</title>
    </item>
    <imsss:sequencing>
      <imsss:rollupRules >
        <imsss:rollupRule childActivitySet = "all">
          <imsss:rollupConditions>
            <imsss:rollupCondition condition = "attempted"/>
          </imsss:rollupConditions>
          <imsss:rollupAction action = "completed"/>
        </imsss:rollupRule>
      </imsss:rollupRules>
    </imsss:sequencing>
  </item>
</item>
```

コード例5-11

5.1.6.1. <rollupRule>要素

<rollupRule>要素は、アクティビティに適用される各ロールアップルールのコンテナである。ルールの一般的な形式はインフォーマルに “ If child activity set, condition set Then action ” と表現される。複数の条件が許される。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <rollupRule>

データタイプ: <rollupRule>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<rollupRule>要素は以下の要素/属性を含む：

属性:

- childActivitySet (オプション。デフォルト値 = all) 。この属性は、どの（訳注：どの子アクティビティの）データ値がロールアップ条件の評価に用いられるかを示す[5]。 XML Data Type: xs:token. 以下が使うことのできる語彙のリストである：
 - o all
 - o any

- o none
- o atLeastCount
- o atLeastPercent
- minimumCount (オプション. デフォルト値 = 0) . minimumCount 属性は, childActivitySet 属性が atLeastCount に設定されたときに使用される. 最低でもこの属性で指定された数の子アクティビティのロールアップ条件が true ならば, ロールアップ条件は true となる[5]. XML Data Type: xs:nonNegativeInteger.
- minimumPercent (オプション. デフォルト値 = 0.0000).. minimumPercent 属性は, childActivitySet 属性が atLeastPercent に設定されたときに使用される. 最低でもこの属性で指定された比率の子アクティビティのロールアップ条件が true ならば, ロールアップ条件は true となる[5]. XML Data Type: xs:decimal (範囲: 0.0000 から 1.0000 . 精度: 最低小数点以下 4 桁).

要素:

- <rollupConditions>
- <rollupAction>

繰り返し: <rollupRules>要素で 0 回存在する.

例:

```
<item identifier="MODULE2">
  <title>Module 2 -- Enhancing Images</title>
  <item identifier="PRETEST2">
    <title>Module 2 -- Pretest</title>
    <item identifier="PRETEST_QUESTION4" isvisible = "false"
      identifierref="RESOURCE_QUESTION4">
      <title>Question 4</title>
    </item>
    <item identifier="PRETEST_QUESTION5" isvisible = "false"
      identifierref="RESOURCE_QUESTION5">
      <title>Question 5</title>
    </item>
    <item identifier="PRETEST_QUESTION6" isvisible = "false"
      identifierref="RESOURCE_QUESTION6">
      <title>Question 6</title>
    </item>
    <imsss:sequencing>
      <imsss:rollupRules >
        <imsss:rollupRule childActivitySet = "all">
          <imsss:rollupConditions>
            <imsss:rollupCondition condition = "attempted"/>
          </imsss:rollupConditions>
          <imsss:rollupAction action = "completed"/>
        </imsss:rollupRule>
      </imsss:rollupRules>
    </imsss:sequencing>
  </item>
</item>
```

コード例5-12

5.1.6.1.1. <rollupConditions>要素

<rollupConditions>要素は, 単一ロールアップルール内で適用される条件の集合のコンテナである.

XML 名前空間: `http://www.imsglobal.org/xsd/imsss`

XML 名前空間接頭辞: `imsss`

XML バインディング表現: `<rollupConditions>`

データタイプ: `<rollupConditions>`要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。`<rollupConditions>`要素は以下の要素/属性を含む:

属性:

- `conditionCombination` (オプション。デフォルト値 = `any`)。この属性は、どのようにロールアップ条件が組み合わされるかを示す。XML Data Type: `xs:token`。この属性は、以下のトークンの内一つの値を含む:
 - `all`
 - `any`

要素:

- `<rollupCondition>`

繰り返し: `<rollupRule>`要素で1回存在する。

例:

```
<item identifier="MODULE2">
  <title>Module 2 -- Enhancing Images</title>
  <item identifier="PRETEST2">
    <title>Module 2 -- Pretest</title>
    <item identifier="PRETEST_QUESTION4" isvisible = "false"
      identifierref="RESOURCE_QUESTION4">
      <title>Question 4</title>
    </item>
    <item identifier="PRETEST_QUESTION5" isvisible = "false"
      identifierref="RESOURCE_QUESTION5">
      <title>Question 5</title>
    </item>
    <item identifier="PRETEST_QUESTION6" isvisible = "false"
      identifierref="RESOURCE_QUESTION6">
      <title>Question 6</title>
    </item>
    <imsss:sequencing>
      <imsss:rollupRules >
        <imsss:rollupRule childActivitySet = "all">
          <imsss:rollupConditions>
            <imsss:rollupCondition condition = "attempted"/>
          </imsss:rollupConditions>
          <imsss:rollupAction action = "completed"/>
        </imsss:rollupRule>
      </imsss:rollupRules>
    </imsss:sequencing>
  </item>
</item>
```

コード例5-13

5.1.6.1.2. <rollupCondition>要素

<rollupCondition>要素は、ロールアップルールで適用される条件を指定する[5].

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <rollupCondition>

データタイプ: <rollupCondition>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<rollupCondition>要素は以下の要素/属性を含む:

属性:

- operator (オプション。デフォルト値 = noOp)。個別条件に適用される単項論理演算子[5]。XML Data Type: xs:token。この属性は、以下のトークンの内一つの値を含む:
 - not
 - noOp
- condition (必須) ルールの条件要素を示す。XML Data Type:xs:token。以下のトークンの内一つの値を含む:
 - satisfied
 - objectiveStatusKnown
 - objectiveMeasureKnown
 - completed
 - activityProgressKnown
 - attempted
 - attemptLimitExceeded
 - timeLimitExceeded
 - outsideAvailableTimeRange

要素:

- なし

繰り返し: <rollupConditions>要素で1回以上存在する。

例：

```
<item identifier="MODULE2">
  <title>Module 2 -- Enhancing Images</title>
  <item identifier="PRETEST2">
    <title>Module 2 -- Pretest</title>
    <item identifier="PRETEST_QUESTION4" isvisible = "false"
      identifierref="RESOURCE_QUESTION4">
      <title>Question 4</title>
    </item>
    <item identifier="PRETEST_QUESTION5" isvisible = "false"
      identifierref="RESOURCE_QUESTION5">
      <title>Question 5</title>
    </item>
    <item identifier="PRETEST_QUESTION6" isvisible = "false"
      identifierref="RESOURCE_QUESTION6">
      <title>Question 6</title>
    </item>
    <imsss:sequencing>
      <imsss:rollupRules >
        <imsss:rollupRule childActivitySet = "all">
          <imsss:rollupConditions>
            <imsss:rollupCondition condition = "attempted"/>
          </imsss:rollupConditions>
          <imsss:rollupAction action = "completed"/>
        </imsss:rollupRule>
      </imsss:rollupRules>
    </imsss:sequencing>
  </item>
</item>
```

コード例 5-14

5.1.6.1.3. <rollupAction>要素

<rollupAction>要素は、ロールアップルールで適用される条件（訳注：動作）を指定する[5].

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <rollupAction>

データタイプ: <rollupAction>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<rollupAction>要素は以下の要素/属性を含む：

属性:

- **action**（必須）。この属性は、ルールが true と評価された場合に求められるロールアップ動作を示す。XML Data Type: xs:token。この属性は、以下のトークンの内一つの値を含む：
 - o **satisfied**
 - o **notSatisfied**
 - o **completed**
 - o **incomplete**

要素:

- なし

繰り返し: <rollupRule>要素で1回存在する.

例:

```
<item identifier="MODULE2">
  <title>Module 2 -- Enhancing Images</title>
  <item identifier="PRETEST2">
    <title>Module 2 -- Pretest</title>
    <item identifier="PRETEST_QUESTION4" isvisible = "false"
      identifierref="RESOURCE_QUESTION4">
      <title>Question 4</title>
    </item>
    <item identifier="PRETEST_QUESTION5" isvisible = "false"
      identifierref="RESOURCE_QUESTION5">
      <title>Question 5</title>
    </item>
    <item identifier="PRETEST_QUESTION6" isvisible = "false"
      identifierref="RESOURCE_QUESTION6">
      <title>Question 6</title>
    </item>
    <imsss:sequencing>
      <imsss:rollupRules >
        <imsss:rollupRule childActivitySet = "all">
          <imsss:rollupConditions>
            <imsss:rollupCondition condition = "attempted"/>
          </imsss:rollupConditions>
          <imsss:rollupAction action = "completed"/>
        </imsss:rollupRule>
      </imsss:rollupRules>
    </imsss:sequencing>
  </item>
</item>
```

コード例5-15

5.1.7. <objectives>要素

<objectives>要素は、アクティビティに関連付けられた学習目標の集合のコンテナである[5]。各アクティビティは、最低一つの主学習目標を持たなければならず、無制限の個数の学習目標を持つことが可能である。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <objectives>

データタイプ: <objectives>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<objectives>要素は以下の要素/属性を含む:

属性:

- なし

要素:

- <primaryObjective>
- <objective>

繰り返し: <sequencing>要素で 0 回か 1 回存在する。

例:

```
<item identifier="POSTTEST1">
  <title>Module 1 -- Posttest</title>
  <item identifier="POSTTEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="POSTTEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <item identifier="POSTTEST_QUESTION3" isvisible = "false"
    identifierref="RESOURCE_QUESTION3">
    <title>Question 3</title>
  </item>
  <imsss:sequencing>
    <imsss:objectives>
      <imsss:primaryObjective objectiveID = "PRIMARYOBJ"
        satisfiedByMeasure = "true">
        <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
        <imsss:mapInfo targetObjectiveID = "obj_module_1"
          readNormalizedMeasure = "false"
          writeSatisfiedStatus = "true" />
      </imsss:primaryObjective>
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

コード例5-16

5.1.7.1. <primaryObjective>要素

<primaryObjective>要素は、アクティビティに関連付けられた、ロールアップに関与する学習目標を指定する[5]。<objectives>要素が定義された場合、<primaryObjective>は必須である(しかし、要素は、空の文字列として表現されることがある - <primaryObjective/>)。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <primaryObjective>

データタイプ: <primaryObjective>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<primaryObjective>要素は以下の要素/属性を含む:

属性:

- **satisfiedByMeasure** (オプション。デフォルト値 = false)。この属性は(値が true に設定された場合)、アクティビティに関連付けられた学習目標の習得状態を決定するために、他の方法ではなく、<minNormalizedMeasure>を使用することを示す[5]。XML Data Type: xs:boolean.
- **objectiveID** (オプション)。アクティビティに関連付けられた学習目標の識別子[5]。objectiveID のデータタイプは、IMS シンプルシーケンシング仕様で定義された通り、一意の識別子である。空の文字列は、どの目標が参照されるのか一意に特定するために十分な意味上の情報を提供しないので、空文字列の objectiveID 属性は許されず、すべてブランクの文字(XML パーサが空文字列に置き換える文字)を含むことはできない。

<primaryObjective>が学習目標マップ(<mapInfo>)を含む場合、objectiveID 属性が必要となる。<primaryObjective>が目標マップを含まない場合、objectiveID 属性はオプションである。

アクティビティに対して定義された学習目標のセットに対して(例えば、ある<objectives>要素でのひとつの<primaryObjective>と複数の<objective>要素)、全ての定義された objectiveID 属性は一意でなくてはならない。LMS は、cmi.objectives.n.id データモデル要素を初期化するために、objectiveID 属性に保持された値を用いる。XML Data Type: xs:anyURI.

要素:

- <minNormalizedMeasure>
- <mapInfo>

繰り返し: <objectives>要素で 1 回だけ存在する。

例：

```
<item identifier="POSTTEST1">
  <title>Module 1 -- Posttest</title>
  <item identifier="POSTTEST_QUESTION1" isVisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="POSTTEST_QUESTION2" isVisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <item identifier="POSTTEST_QUESTION3" isVisible = "false"
    identifierref="RESOURCE_QUESTION3">
    <title>Question 3</title>
  </item>
  <imsss:sequencing>
    <imsss:objectives>
      <imsss:primaryObjective objectiveID = "PRIMARYOBJ"
        satisfiedByMeasure = "true">
        <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
        <imsss:mapInfo targetObjectiveID = "obj_module_1"
          readNormalizedMeasure = "false"
          writeSatisfiedStatus = "true" />
        </imsss:primaryObjective>
      </imsss:objectives>
    </imsss:sequencing>
  </item>
```

コード例5-17

5.1.7.1.1. <minNormalizedMeasure> 要素

<minNormalizedMeasure>要素は、学習目標に対する最小習得度を指定する[5]。値は -1 から 1 の間で正規化される。主学習目標 (<primaryObjective>要素) の satisfiedByMeasure が真の場合、LMS はこの値を用いて cmi.scaled_passing_score (SCORM RTE ブック参照[2]) を初期化する。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <minNormalizedMeasure>

データタイプ: <minNormalizedMeasure>要素の値は xs:decimal タイプである。値が提供されない場合、デフォルト値は 1.0 である。

繰り返し: <primaryObjective>および<objective>要素で 0 回か 1 回存在する。

例：

```
<item identifier="POSTTEST1">
  <title>Module 1 -- Posttest</title>
  <item identifier="POSTTEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="POSTTEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <item identifier="POSTTEST_QUESTION3" isvisible = "false"
    identifierref="RESOURCE_QUESTION3">
    <title>Question 3</title>
  </item>
  <imsss:sequencing>
    <imsss:objectives>
      <imsss:primaryObjective objectiveID = "PRIMARYOBJ"
        satisfiedByMeasure = "true">
        <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
        <imsss:mapInfo targetObjectiveID = "obj_module_1"
          readNormalizedMeasure = "false"
          writeSatisfiedStatus = "true" />
      </imsss:primaryObjective>
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

コード例5-18

5.1.7.1.2. <mapInfo>要素

<mapInfo>要素は学習目標マップ記述のコンテナである。これは、アクティビティのローカル学習目標情報と共有グローバル学習目標とのマッピングを定義する。各アクティビティが持つことができる目標マップの数には制限がない。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <mapInfo>

データタイプ: <mapInfo>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<mapInfo>要素は以下の要素/属性を含む：

属性:

- **targetObjectiveID** (必須)。マッピングの対象のグローバル共有学習目標 の識別子 [5]。 targetObjectiveID のデータタイプは、IMS シンプルシーケンシング仕様で定義された通り、一意の識別子である。空の文字列は、どのグローバル共有学習目標が対象とされるか一意に特定するのに十分な情報を提供しないので、空文字列の targetObjectiveID 属性は許されず、すべてブランクの文字 (XML パーサが空文字列に置き換える文字) を含むことはできない。XML Data Type: xs:anyURI。
- **readSatisfiedStatus** (オプション。デフォルト値 = true)。この属性は、指定されたローカル学習目標の進捗状況が定義されていないとき、ローカル学習目標の習得状態は、

指定された共有グローバル学習目標から取得することを示す[5]. XML Data Type: xs:boolean.

- readNormalizedMeasure (オプション. デフォルト値 = true). この属性は, 指定されたローカル学習目標に対する正規化習得度が定義されていないとき, ローカル学習目標に対する正規化習得度は, 指定された共有グローバル学習目標から取得することを示す[5]. XML Data Type: xs:boolean.
- writeSatisfiedStatus (オプション. デフォルト値 = false). この属性は, アクティビティへの試行終了(Termination(" "))時に, 指定されたローカル学習目標の習得状態が, 指定された共有グローバル学習目標に転送されることを示す[5]. XML Data Type: xs:boolean.
- writeNormalizedMeasure (オプション. デフォルト値 = false). この属性は, アクティビティへの試行終了(Termination(" "))時に, 指定されたローカル学習目標の正規化習得度が, 指定された共有グローバル学習目標に転送されることを示す[5]. XML Data Type: xs:boolean.

要素:

- なし

繰り返し: <primaryObjective> and <objective>要素で 0 回以上存在する.

読み出し学習目標マップ (readSatisfiedStatus および readNormalizedMeasure) に対して:

- 学習目標 (<primaryObjective>もしくは<objective>) に対して, 複数の<mapInfo>要素が存在する場合, readSatisfiedStatus 属性が"true"の<mapInfo>要素は一つだけでなくてはならない.
- 学習目標 (<primaryObjective>もしくは<objective>) に対して, 複数の<mapInfo>要素が存在する場合, readNormalizedMeasure 属性が"true"の<mapInfo>要素は一つだけでなくてはならない.

書き込み学習目標マップ (writeSatisfiedStatus および writeNormalizedMeasure) に対して:

- アクティビティに対して, 複数の学習目標 (<primaryObjective>もしくは<objective>) の<mapInfo>要素が同一の targetObjectiveID を共有する場合, writeSatisfiedStatus 属性が"true"の学習目標は一つだけでなくてはならない.
- アクティビティに対して, 複数の学習目標 (<primaryObjective>もしくは<objective>) の<mapInfo>要素が同一の targetObjectiveID を共有する場合, writeNormalizedMeasure 属性が"true"の学習目標は一つだけでなくてはならない.

例：

```
<item identifier="POSTTEST1">
  <title>Module 1 -- Posttest</title>
  <item identifier="POSTTEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="POSTTEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <item identifier="POSTTEST_QUESTION3" isvisible = "false"
    identifierref="RESOURCE_QUESTION3">
    <title>Question 3</title>
  </item>
  <imsss:sequencing>
    <imsss:objectives>
      <imsss:primaryObjective objectiveID = "PRIMARYOBJ"
        satisfiedByMeasure = "true">
        <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
        <imsss:mapInfo targetObjectiveID = "obj_module_1"
          readNormalizedMeasure = "false"
          writeSatisfiedStatus = "true" />
      </imsss:primaryObjective>
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

コード例5-19

5.1.7.2. <objective>要素

<objective>要素は、アクティビティに関連付けられた、ロールアップに関与しない学習目標を指定する[5]。この要素は、<primaryObjective>が定義されている場合にだけ存在しうる。

XML 名前空間： <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞： imsss

XML バインディング表現： <objective>

データタイプ： <objective>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<objective>要素は以下の要素/属性を含む：

属性：

- **satisfiedByMeasure**（オプション、デフォルト値 = false）。この属性は（値が true に設定された場合）、アクティビティに関連付けられた学習目標の習得状態を決定するために、他の方法ではなく、<minNormalizedMeasure>を使用することを示す[5]。XML Data Type: xs:boolean.
- **objectiveID**（オプション）。アクティビティに関連付けられた学習目標の識別子[5]。objectiveID のデータタイプは、IMS シンプルシーケンシング仕様で定義された通り、一意の識別子である。空の文字列は、どの目標が参照されるのか一意に特定するために十分な意味上の情報を提供しないので、空文字列の objectiveID 属性は許されず、すべてブランクの文字（XML パーサが空文字列に置き換える文字）を含むことはできない。

アクティビティに対して定義された学習目標のセットに対して（例えば、ある <objectives>要素での ひとつの<primaryObjective> と複数の <objective>要素）、全ての定義された objectiveID 属性は一意でなくてはならない。LMS は、cmi.objectives.n.id データモデル要素を初期化するために、objectiveID 属性に保持された値を用いる。XML Data Type: xs:anyURI.

要素:

- <minNormalizedMeasure> (「5.1.7.1.1 <minNormalizedMeasure>」参照)
- <mapInfo> (「5.1.7.1.2 <mapInfo>」参照)

繰り返し: <objectives>要素で 0 回存在する。

例:

```
<item identifier="POSTTEST1">
  <title>Module 1 -- Posttest</title>
  <item identifier="POSTTEST_QUESTION1" isVisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="POSTTEST_QUESTION2" isVisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <item identifier="POSTTEST_QUESTION3" isVisible = "false"
    identifierref="RESOURCE_QUESTION3">
    <title>Question 3</title>
  </item>
  <imsss:sequencing>
    <imsss:objectives>
      <imsss:primaryObjective objectiveID = "PRIMARYOBJ" />
      <imsss:objective objectiveID="obj_module_1">
        <imsss:mapInfo targetObjectiveID="obj_module_1"
          readSatisfiedStatus = "false"
          readNormalizedMeasure = "false"
          writeSatisfiedStatus = "true" />
      </imsss:objective>
    </imsss:objectives>
  </imsss:sequencing>
</item>
```

コード例5-20

5.1.8. <randomizationControls>要素

<randomizationControls>要素は、シーケンスプロセス中にアクティビティの子をどのように順序付けるかの記述のためのコンテナである。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <randomizationControls>

データタイプ: <randomizationControls>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。

<randomizationControls>要素は以下の要素/属性を含む：

属性:

- randomizationTiming (オプション。デフォルト値 = never)。この属性は、いつアクティビティの子の順番付けを行うかを示す[5]。XML Data Type: xs:token。この属性は、以下のトークンの内一つの値を含む：
 - o never
 - o once
 - o onEachNewAttempt
- selectCount (オプション。デフォルト値 = 0)。この属性は、アクティビティに関連付けられた子アクティビティから選択されなければならない子アクティビティの数を示す[5]。XML Data Type: xs:nonNegativeInteger。
- reorderChildren (オプション。デフォルト値 = false)。この属性は、子アクティビティの順序がランダム化されている（訳注：子アクティビティの順序をランダム化する）ことを示す[5]。XML Data Type: xs:boolean。
- selectionTiming (オプション。デフォルト値 = never)。この属性は、いつ（訳注：子アクティビティの）選択を行うかを示す。XML Data Type: xs:token。この属性は以下のトークンの内一つの値を含む：
 - o never
 - o once
 - o onEachNewAttempt

要素:

- なし

繰り返し: <sequencing>要素で 0 回か 1 回存在する。

例：

```
<item identifier="POSTTEST1">
  <title>Module 1 -- Posttest</title>
  <item identifier="POSTTEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="POSTTEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <item identifier="POSTTEST_QUESTION3" isvisible = "false"
    identifierref="RESOURCE_QUESTION3">
    <title>Question 3</title>
  </item>
  <imsss:sequencing>
    <imsss:randomizationControls selectCount="2"
      selectionTiming="onEachNewAttempt" />
  </imsss:sequencing>
</item>
```

コード例5-21

5.1.9. <deliveryControls> 要素

<deliveryControls>要素は、シーケンスプロセス中にアクティビティの子をどのように順序付けるか（訳注：アクティビティを配信する際の属性）の記述のためのコンテナである。

XML 名前空間: <http://www.imsglobal.org/xsd/imsss>

XML 名前空間接頭辞: imsss

XML バインディング表現: <deliveryControls>

データタイプ: <deliveryControls>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<deliveryControls>要素は以下の要素/属性を含む：

属性:

- **tracked**（オプション。デフォルト値 = true）。この属性は、試行における学習目標進捗情報およびアクティビティ/アテンプト進捗情報が記録され、このデータが親アクティビティのロールアップに関与することを示す[5]。XML Data Type: xs:boolean.
- **completionSetByContent**（オプション。デフォルト値 = false）。この属性は、アクティビティの試行完結状態が、SCOによって設定されることを示す[5]。XML Data Type: xs:boolean.
- **objectiveSetByContent**（オプション。デフォルト値 = false）。この属性は、アクティビティに関連づけられたロールアップに寄与する学習目標の学習目標習得状態が、SCOによって設定されることを示す。XML Data Type: xs:boolean.

要素:

- なし

繰り返し: <sequencing>要素で 0 回か 1 回存在する。

例:

```
<item identifier="LESSON1" identifierref="RESOURCE_LESSON1">
  <title>Lesson 1 -- Interface</title>
  <imsss:sequencing>
    <imsss:deliveryControls tracked = "false"/>
  </imsss:sequencing>
</item>
```

コード例5-22

5.1.10. <constrainedChoiceConsiderations> 要素

<adlseq:constrainedChoiceConsiderations>要素は、シーケンシングプロセス中に、どのように choice ナビゲーション要求を制約するかの記事のためのコンテナである。constrained choice は、それが定義されたアクティビティにだけ適用される。

XML 名前空間: http://www.adlnet.org/xsd/adlseq_v1p3

XML 名前空間接頭辞: adlseq

XML バインディング表現: <adlseq:constrainedChoiceConsiderations>

データタイプ: <adlseq:constrainedChoiceConsiderations>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。

<adlseq:constrainedChoiceConsiderations>要素は以下の要素/属性を含む：

属性:

- preventActivation (オプション。デフォルト値 = false)。この属性は、現在のアクティビティが親でない限り、子アクティビティへの試行を開始できないことを示す。XML Data Type: xs:boolean.
- constrainChoice (オプション。デフォルト値 = false)。この属性は、制約されたアクティビティの論理的に次のアクティビティだけが、choice ナビゲーション要求の対象になりうることを示す。XML Data Type: xs:boolean.

要素:

- なし

繰り返し: <sequencing>要素で 0 回か 1 回存在する。

例：

```
<item identifier="Module1">
  <item identifier="EXAM1">
    <title>Module 1 -- Exam</title>
    <item identifier="QUESTION1" isvisible = "false"
      identifierref="RESOURCE_QUESTION1">
      <title>Question 1</title>
    </item>
    <item identifier="QUESTION2" isvisible = "false"
      identifierref="RESOURCE_QUESTION2">
      <title>Question 2</title>
    </item>
    <item identifier="QUESTION3" isvisible = "false"
      identifierref="RESOURCE_QUESTION3">
      <title>Question 3</title>
    </item>
    <imsss:sequencing>
      <imsss:controlMode choice="false" choiceExit ="false" flow="true"
        forwardOnly="true"/>
      <imsss:rollupRules>
        <imsss:rollupRule childActivitySet="all">
          <imsss:rollupConditions>
            <imsss:rollupCondition condition="attempted"/>
          </imsss:rollupConditions>
          <imsss:rollupAction action="completed"/>
        </imsss:rollupRule>
      </imsss:rollupRules>
      <imsss:objectives>
        <imsss:primaryObjective satisfiedByMeasure="true">
          <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
        </imsss:primaryObjective>
      </imsss:objectives>
    </imsss:sequencing>
  </item>
  <imsss:sequencing>
    <imsss:controlMode flow = "true"/>
    <imsss:sequencingRules>
      <imsss:exitConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "completed"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "exit"/>
      </imsss:exitConditionRule>
    </imsss:sequencingRules>
    <adlseq:constrainedChoiceConsiderations constrainChoice = "true" />
  </imsss:sequencing>
</item>
```

コード例5-23

5.1.11. <rollupConsiderations> 要素

<adlseq:rollupConsiderations>要素は、どのような場合にアクティビティがロールアッププロセスに含まれるのかを記述するためのコンテナである。

XML 名前空間: http://www.adlnet.org/xsd/adlseq_v1p3

XML 名前空間接頭辞: adlseq

XML バインディング表現: <rollupConsiderations>

データタイプ: <rollupConsiderations>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。

<rollupConsiderations>要素は以下の要素/属性を含む:

属性:

- **requiredForSatisfied** (オプション, デフォルト値 = always) . この属性は, 親の Satisfied ロールアップルールの評価にアクティビティが含まれる条件を示す. XML Data Type: xs:token.
- **requiredForNotSatisfied** (オプション, デフォルト値 = always) . この属性は, 親の Not Satisfied ロールアップルールの評価にアクティビティが含まれる条件を示す. XML Data Type: xs:token.
- **requiredForCompleted** (オプション, デフォルト値 = always) . この属性は, 親の Completed ロールアップルールの評価にアクティビティが含まれる条件を示す. XML Data Type: xs:token.
- **requiredForIncomplete** (オプション, デフォルト値 = always) . この属性は, 親の Incomplete ロールアップルールの評価にアクティビティが含まれる条件を示す. XML Data Type: xs:token.
- **measureSatisfactionIfActive** (オプション, デフォルト値 = false) .. この属性はアクティビティがアクティブなとき, ロールアップで習得状態を決定するのに学習目標習得度を用いるかどうかを示す. XML Data Type: xs:boolean.

上記の各々の属性は以下の制限されたトークンの内一つの値を持つ:

- **always** (デフォルト): アクティビティは常にロールアップルール処理に含まれる
- **ifAttempted**: アクティビティは, アクティビティが試行された場合, ロールアッププロセスに含まれる
- **ifNotSkipped**: アクティビティは, アクティビティがスキップされなかった場合, ロールアッププロセスに含まれる
- **ifNotSuspended**: アクティビティは, アクティビティが中断されなかった場合, ロールアッププロセスに含まれる

要素:

- なし

繰り返し: <sequencing>要素で 0 回か 1 回存在する。

例：

```
<item identifier="POSTTEST3">
  <title>Module 3 -- Posttest</title>
  <item identifier="POSTTEST_QUESTION7" isvisible = "false"
    identifierref="RESOURCE_QUESTION7">
    <title>Question 7</title>
  </item>
  <item identifier="POSTTEST_QUESTION8" isvisible = "false"
    identifierref="RESOURCE_QUESTION8">
    <title>Question 8</title>
  </item>
  <item identifier="POSTTEST_QUESTION9" isvisible = "false"
    identifierref="RESOURCE_QUESTION9">
    <title>Question 9</title>
  </item>
  <imsss:sequencing>
    <imsss:controlMode choice = "false" choiceExit = "false" flow = "true"
      forwardOnly = "true"/>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "always"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "hiddenFromChoice"/>
      </imsss:preConditionRule>
      <adlseq:rollupConsiderations measureSatisfactionIfActive = "false"
        requiredForCompleted = "ifNotSkipped" />
    </imsss:sequencingRules>
  </imsss:sequencing>
</item>
```

コード例5-24

5.1.12. <sequencingCollection>要素

シーケンシング情報（<sequencing>要素の中で定義された情報）の集合が、再利用できる場合がある。IMS SS 仕様の XML Binding は、シーケンシング情報の集合のコンテナの役割を果たす <sequencingCollection>要素を定義する。<sequencing>要素の IDRef 属性が、<sequencingCollection>の子要素である<sequencing>要素の ID 属性を参照するときに、再利用が行われる。<sequencing>要素が IDRef 属性およびシーケンシング情報のインライン定義の両方を用いた場合、インラインで定義されたトップレベル要素が、参照された要素で定義された同様の要素よりも優先する。

シーケンシング集合の特徴と要件で重要なものは以下の通りである。

1. <sequencingCollection>を用いる場合、シーケンシング集合は一つだけ存在する。
2. シーケンシング集合中で定義された<sequencing>要素を参照するように<sequencing>要素を設定する場合、IDRef 属性は必須であり、それはシーケンシング集合中で定義された<sequencing>要素で指定された “ID” を参照する。
3. <sequencing>要素が<item>もしくは<organization>要素の子として定義された場合、ID 属性は許されない。
4. シーケンシング集合中で定義されたすべての<sequencing>要素において、ID 属性は必須であり、IDRef 属性は許されない。この要件はシーケンシング情報のチェーン化（他のシーケンシング情報を参照するシーケンシング情報）を防ぐためにおかれている。この要件は将来緩和される可能性があるが、シーケンシング情報のチェーン化の利用に対するより多くのユースケースが必要である。

属性:

- なし

要素:

- <sequencingCollection>がシーケンシング情報を定義するのに利用されると <sequencing>要素が 1 回以上存在する。

繰り返し: <sequencingCollection>要素は、<imscp:manifest>要素の子として 0 回か 1 回存在する。

例：

```
<item identifier="PRETEST1">
  <title>Module 1 -- Pretest</title>
  <item identifier="PRETEST_QUESTION1" isvisible = "false"
    identifierref="RESOURCE_QUESTION1">
    <title>Question 1</title>
  </item>
  <item identifier="PRETEST_QUESTION2" isvisible = "false"
    identifierref="RESOURCE_QUESTION2">
    <title>Question 2</title>
  </item>
  <item identifier="PRETEST_QUESTION3" isvisible = "false"
    identifierref="RESOURCE_QUESTION3">
    <title>Question 3</title>
  </item>
  <imsss:sequencing IDRef = "pretest">
    <imsss:objectives>
      <imsss:primaryObjective objectiveID = "PRIMARYOBJ"
        satisfiedByMeasure = "true">
        <imsss:minNormalizedMeasure>0.6</imsss:minNormalizedMeasure>
        <imsss:mapInfo targetObjectiveID = "obj_module_1"
          readNormalizedMeasure = "false"
          writeSatisfiedStatus = "true"
          writeNormalizedMeasure = "true"  />
        </imsss:primaryObjective>
      </imsss:objectives>
    </imsss:sequencing>
  </item>

<!-- Other Manifest data -->

<imsss:sequencingCollection>
  <imsss:sequencing ID = "pretest">
    <imsss:controlMode choice = "false" choiceExit = "false" flow = "true"
      forwardOnly = "true"/>
    <imsss:sequencingRules>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "completed"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "disabled"/>
      </imsss:preConditionRule>
      <imsss:preConditionRule>
        <imsss:ruleConditions>
          <imsss:ruleCondition condition = "satisfied"/>
        </imsss:ruleConditions>
        <imsss:ruleAction action = "skip"/>
      </imsss:preConditionRule>
    </imsss:sequencingRules>
  </imsss:sequencing>
</imsss:sequencingCollection>
```

コード例5-25

5.2. プレゼンテーション/ナビゲーション情報

SCORM は、IMS SS 仕様と調和したプレゼンテーション/ナビゲーションガイダンスを定義する。ADL は、プレゼンテーションとナビゲーションに関して ADL コミュニティからユースケースおよび要件を収集していく計画である。

プレゼンテーション/ナビゲーション情報の XML Binding はコンテンツパッケージングマニフェスト XML スキーマへの拡張によって取り扱われる。<adlnav:presentation>と呼ばれる新たな要素が仕様化されている。<adlnav:presentation>要素は、<navigationInterface>と呼ばれる単一のサブ要素を含む。<adlnav:navigationInterface>要素は、<adlnav:hideLMSUI>と呼ばれるサブ要素を持つ。

5.2.1.1. <presentation>要素

<presentation>要素は、ある特定の学習アクティビティのプレゼンテーション情報を含むコンテナである。この要素は、IMS コンテンツパッケージング仕様に対する ADL が定義した拡張である。この要素は、必要な場合、SCO を参照するリーフ<item>要素の子としてだけ現れる。

XML 名前空間: http://www.adlnet.org/xsd/adlnav_v1p3

XML 名前空間接頭辞: adlnav

XML バインディング表現: <presentation>

データタイプ: <presentation>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。<presentation>要素は以下の要素/属性を含む：

属性:

- なし

要素:

- <navigationInterface>

繰り返し: <presentation>要素は、<imscp:item>要素内で 0 回か 1 存在する。

例:

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
    <title>Content 1</title>
    <adlnav:presentation>
      <adlnav:navigationInterface>
        <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
        <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
      </adlnav:navigationInterface>
    </adlnav:presentation>
  </item>
</organization>
```

コード例 5-26

5.2.1.1.1. <navigationInterface>要素

<navigationInterface>要素は、ある特定の学習アクティビティに対するナビゲーションインターフェースプレゼンテーション要件を含むコンテナ要素である。

XML 名前空間: http://www.adlnet.org/xsd/adlnav_v1p3

XML 名前空間接頭辞: adlnav

XML バインディング表現: <navigationInterface>

データタイプ: <navigationInterface>要素は親要素である。親要素は自身に関連付けられた値を持たない。親要素は他の要素に対してコンテナとしての役割を果たす。
<navigationInterface>要素は以下の要素/属性を含む：

属性:

- なし

要素:

- <hideLMSUI>

繰り返し: <navigationInterface>要素は、<presentation>要素内で 0 回か 1 回存在する。

例:

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
    <title>Content 1</title>
    <adlnav:presentation>
      <adlnav:navigationInterface>
        <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
        <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
      </adlnav:navigationInterface>
    </adlnav:presentation>
  </item>
</organization>
```

コード例5-27

5.2.1.1.1.1. <hideLMSUI>要素

<hideLMSUI>要素は、学習者が特定のイベントを起動するユーザインターフェース装置を、LMS が提供してはならないことを示す。これらのイベントは Previous, Continue, Exit, Exit All, Abandon, Abandon All および Suspend All である。

XML 名前空間: http://www.adlnet.org/xsd/adlnav_v1p3

XML 名前空間接頭辞: adlnav

XML バインディング表現: <hideLMSUI>

データタイプ: <hideLMSUI>要素は xs:string で表現される。文字列は制限された語彙トークンである。デフォルトで、値が含まれていないと、LMS は以下のナビゲーション要求を受け付けるユーザインターフェース装置を提供する。トークンは以下うちの一つである：

- **previous:** 指定された場合、このアクティビティがカレントアクティビティのとき、LMS は“ Previous ”ナビゲーション装置を表示してはならない。
- **continue:** 指定された場合、このアクティビティがカレントアクティビティのとき、LMS は“ Continue ”ナビゲーション装置を表示してはならない。
- **exit:** 指定された場合、このアクティビティがカレントアクティビティのとき、LMS は“ Exit ”ナビゲーション装置を表示してはならない。
- **exitAll:** 指定された場合、このアクティビティがカレントアクティビティのとき、LMS は“ exitAll ”ナビゲーション装置を表示してはならない。
- **abandon:** 指定された場合、このアクティビティがカレントアクティビティのとき、LMS は“ Abandon ”ナビゲーション装置を表示してはならない。
- **abandonAll:** 指定された場合、このアクティビティがカレントアクティビティのとき、LMS は“ AbandonAll ”ナビゲーション装置を表示してはならない。
- **suspendAll:** 指定された場合、このアクティビティがカレントアクティビティのとき、LMS は“ suspendAll ”ナビゲーション装置を表示してはならない。

属性:

- なし

要素:

- なし

繰り返し: <hideLMSUI>要素は、<navigationInterface>要素内で 0 回以上存在する。

例:

```
<organization>
  <item identifier="ITEM3" identifierref="RESOURCE3" isvisible="true">
    <title>Content 1</title>
    <adlnav:presentation>
      <adlnav:navigationInterface>
        <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
        <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
      </adlnav:navigationInterface>
    </adlnav:presentation>
  </item>
</organization>
```

コード例5-28

5.3. コンテンツパッケージングとの関係

IMS コンテンツパッケージング仕様は、学習アクティビティをコンテンツリソースに関係付ける構造、すなわち、<imscp:item>要素と<imscp:resource>要素との関係付けを提供する。さらに、<imscp:item>要素は集合にクラスタ化される。学習アクティビティが親アクティビティにクラスタ化されるため、そのような集合は親の<imscp:organization>要素に包含される。従って IMS SS は、コンテンツパッケージング仕様で定義されたように、学習アクティビティの概念を<imscp:item>要素、<imscp:organization>要素内の<imscp:item>要素の集合、および、<imscp:organization>要素自身にマップする。コンテンツパッケージング XML Binding は、シーケンシング情報がパッケージ化されたコンテンツとどのように関連付けられるかを定義するために、この仕様（訳注：SCORM）によって拡張されている。

学習アクティビティの特定のシーケンスを定義するプロセスは、IMS コンテンツパッケージ仕様の SCORM コンテンツアグリゲーションアプリケーションプロフィールを利用して交換されるコンテンツ集合の作成から始まる。以下の図で示されるように、コンテンツパッケージング<imscp:organization>要素および その中の各<imscp:item>要素は、シーケンシング情報の関連付けによって定義されたシーケンシング動作を保持することができる：

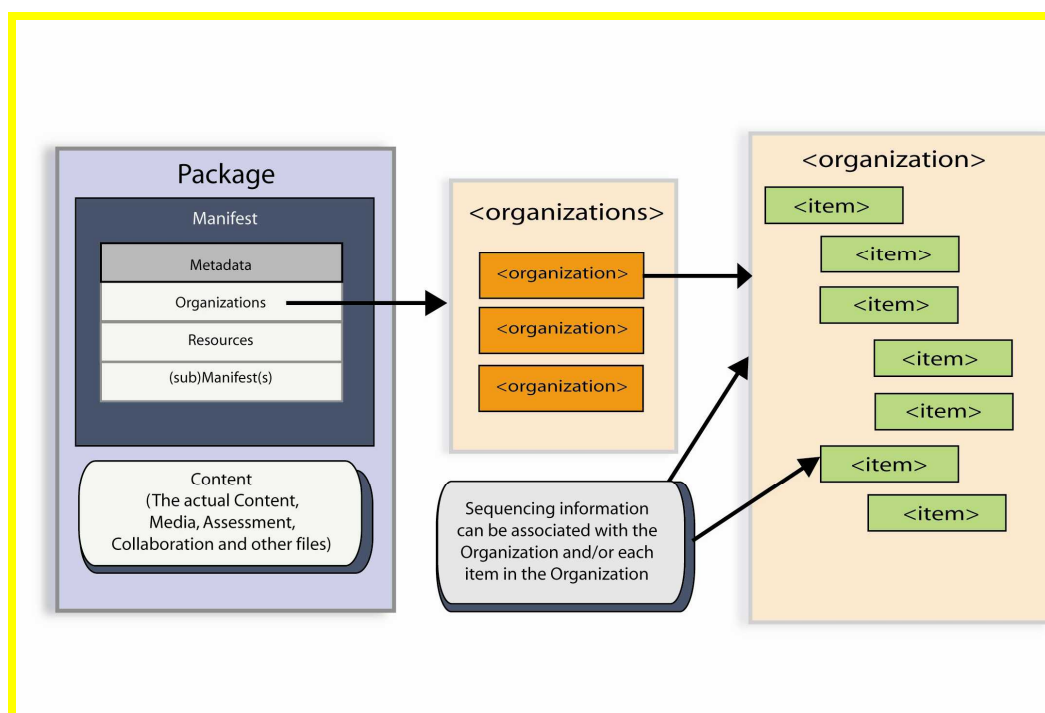


図5.3a: コンテンツパッケージング構成

全ての SCORM 対応コンテンツアグリゲーションパッケージは、デフォルトでシーケンシング情報を含む。SCORM コンテンツパッケージがシーケンシング情報を含まない場合、暗示されたデフォルト動作により、学習者はガイダンスも制約もなしに自由にアクティビティを選択することができる。

このページは空白である。

付録 A 略語表

このページは空白である。

略語表

ADL	Advanced Distributed Learning
AICC	Aviation Industry CBT Committee
API	Application Program Interface
ARIADNE	Alliance of Remote Instructional Authoring & Distribution Networks for Europe
CAM	Content Aggregation Model
CBT	Computer-Based Training
DOI	Digital Object Identifier
HTML	Hypertext Markup Language
IEEE	Institute of Electrical and Electronics Engineers
ISBN	International Standard Book Numbers
ISO	International Standards Organization
ISSN	International Standard Serial Numbers
LMS	Learning Management System
LOM	Learning Objects Metadata
LTSC	Learning Technology Standards Committee
PIF	Package Interchange File
RTE	Run-Time Environment
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SN	Sequencing and Navigation
SPM	Smallest Permitted Maximum
SS	Simple Sequencing
URI	Universal Resource Indicator
URL	Universal Resource Locator
URN	Universal Resource Name
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema Definition

このページは空白である。

付録 B

参考文献

このページは空白である。

参考文献

1. *IEEE 1484.11.2 Standard for Learning Technology – ECMAScript Application Programming Interface for Content to Runtime Services Communication*. November 10, 2003
Available at: <http://www.ieee.org/>
2. *SCORM 2004 3rd Edition Run-Time Environment Version 1.0*, Advanced Distributed Learning, October 20, 2006
Available at: <http://www.adlnet.gov/>
3. *IMS Content Packaging Information Model, Version 1.1.4 Final Specification*. October, 2004
Available at: <http://www.imsglobal.org/>
4. *Aviation Industry CBT Committee (AICC) Computer Managed Instruction Guidelines for Interoperability (CMI001) Version 3.5*. April 2, 2001
Available at: <http://www.aicc.org/>
5. *IMS Simple Sequencing Behavior and Information Model v1.0 Final Specification*, IMS Global Learning Consortium, Inc., March 2003
Available at: <http://www.imsproject.org/>.
6. *Extensible Markup Language (XML) 1.1*. 04 February 2004.
Available at: <http://www.w3.org/>
7. *XML Base*. June 27, 2001
Available at: <http://www.w3.org/>
8. *IETF RFC 3986:2005, Uniform Resource Identifier (URI): Generic Syntax*.
Available at: <http://www.ietf.org/>
9. *IETF RFC 2426:1998, vCard MIME Directory Profile*
10. *ISO/IEC 10646-1:2000, Information technology—Universal multiple-octet coded character set—Part 1: Architecture and basic multilingual plane*.
11. *IEEE 1484.12.1-2002 Learning Object Metadata Standard*.
Available at: <http://www.ieee.org/>
12. *IETF RFC 1951 DEFLATE Compressed Data Format Specification version 1.3*, May 1996
Available at: <http://www.ietf.org/>
13. *XML Schema Part 2: Datatypes*, W3C Recommendation 02 May 2001,
Available at: <http://www.w3.org/>

-
14. *IEEE 1484.12.3 Standard for Extensible Markup Language (XML) Schema Binding for Learning Object Metadata*
Available at: <http://www.ieee.org/>
 15. *IMS Content Packaging Best Practice Guide*, Version 1.1.4 Final Specification, October 2004
Available at: <http://www.imsglobal.org/>
 16. *SCORM 2004 3rd Edition Sequencing and Navigation Version 1.0*, Advanced Distributed Learning, October 20, 2006
Available at: <http://www.adlnet.gov/>

付録 C

ドキュメント改定履歴

このページは空白である。

ドキュメント改定履歴

SCORM バージョン	リリース日付	変更の説明
1.3 ワーキングドラフト 1	2003 年 10 月 22 日	初期ドラフト。変更は以下の通り： - IMS コンテンツパッケージング仕様バージョン 1.1.3 に導入された変更を包含するために SCORM コンテンツパッケージングを更新 - IEEE 1484.12.1-2002 および IEEE 1484.12.3 Draft Standard for Extensible Markup Language (XML) Binding for Learning Object Metadata Data Model の標準化により導入された変更を包含するために SCORM メタデータを更新 - IMS シンプルシーケンシングバージョン 1.0 のサポートを含むために更新 - IMS シンプルシーケンシングバージョン 1.0 を含むためにナビゲーション要件に対するサポートを含める
CAM バージョン 1.3	2004 年 1 月 30 日	更新は以下の通り： - コンテンツアグリゲーション対コンテンツオーガニゼーションの定義の明確化および変更 - SCORM メタデータアプリケーションプロファイルへの名称変更：Package-level Meta-data は Content Aggregation Meta-data へ変更，Content Aggregation Meta-data は Content Organization Meta-data へ変更 - コンテンツオーガニゼーションの定義を明確にするためにセクション 3.3.2 Organizations を更新 - テーブルフォーマットを説明するのにセクション 3.4.1 およびセクション 4.2 に表を追加
CAM バージョン 1.3.1	2004 年 6 月 22 日	更新は以下の通り： - 文法およびスタイルの改善 - アクティビティとリソースを示すために図 2.1.3a を更新 - バリデーション中の well-formedness 問題を解決するためにコード例に多くの更新。属性値に対する正しい引用符の使用の更新。well-formedness エラーを起こした欠けていた要素を追加。well-formedness エラーを起こした要素名称を変更。 - デフォルト属性がマニフェストの識別子を参照で

		きるという記載を削除するために 3.4.1.6 <organizations> 要素を更新. デフォルト属性は <organizations>要素の直接の子孫である <organization>要素だけを参照できる. • 欠けていた<adlcp:completionThreshold>要素および要素の要件を追加 • ローカルリソースは<file>要素を用いて指定しなければならないという説明を追加するためにセクション 3.4.1.23 を更新. • 不正な ID を変更するために図 3.4.2.1a を更新. 元の図は二つのアイテム要素が同一の ID (A002) を持っていた • 表 3.5.3a から漏れていた <adlcp:completionThreshold>要素を追加 • セクション 3.6.1 Packaging Multiple Courses を削除しセクション 3.6.2 Multiple Organizations と交換. (ベストプラクティスができるまで, 複数の個別の“courses”をパッケージ化する理由に関して, より多くの要件収集が必要なため.) • 図 4.1a および対応するテキストの削除 • コンテンツアグリゲーションアプリケーションプロファイルの<metaMetadata>要素に対する繰り返し要件を「0 回もしくは 1 回」から「1 回だけ」に変更. これは SCORM 2004 適合要件バージョン 1.0 で定義された要件および<metadataSchema>子要素に対して定義された要件と一致する • セクション 4.2.8.1 <kind> 要素で誤った語義リファレンスを変更: isbasisof から isbasisfor へ • Updated コンテンツアグリゲーションメタデータアプリケーションプロファイルの要件を更新. メタデータは必須で, 必須の要素は 3.0 Meta-Metadata および Metadata Schema だけである • セクション 5.1.1 <sequencing>要素で見つかった矛盾を更新. セクション 5.1.12 <sequencingCollection>要素にあるものと一致するように繰り返しが「1 回以上」に変更 • セクション 4.4.2.1 Source Declaration および 4.4.2.2 Vocabulary Declarations を削除. このアプローチは LOM を拡張する一つ的手段にすぎない. どのように LOM を拡張するかは SCORM の対象範囲外である. より多くの指針が定式化される必要がある • IMS コンテンツパッケージングに従って, (コンテンツアグリゲーションコンテンツパッケージの) <item>要素に対する要件を「0 回以上」から「1 回以上」へ変更 • セクション 5.1.1: ID および IDRef 属性の利用に対する要件の明確化を追加 • Section 5.1.3.1.1.1: referencedObjective 属性には空
--	--	---

		の文字列もしくは全ブランクは許されないことを明確にするために更新 • Section 5.1.6.1.2: スペルが間違っていた語彙トークンを更新. timeLimitExceeded は“i”が抜けていた • Section 5.1.7: objectiveID 属性には空の文字列もしくは全ブランクは許されないことを示すために更新. <primaryObjective>が学習目標マップを含めば objectiveID が要求されるという要件を定義するために更新. また objectiveID 属性の値は一意でなくてはならないことを示すためにテキストを更新 • Section 5.1.7.1.1: LMS は cmi.scaled_passing_score RTE Data Model Element を初期化するためにこの要素をしなくてはならないことを示すために更新 • Section 5.1.7.1.2: targetObjectiveID 属性には空の文字列ないし全ブランクは許されないこと示すために更新 • Section 5.1.7.1.2: Read Objective Maps および Write Objective Maps の要件に関するより多くの情報, 特に複数学習目標の場合を追加 • Section 5.1.7.2: objectiveID 属性には空の文字列ないし全ブランクは許されないこと示すために更新. また objectiveID 属性の値は一意でなくてはならないことを示すためにテキストを更新 • Randomization Controls の例を示すためにコード例 5-21 を更新
SCORM 2004 第3版 CAM パー ジョン 1.0	2006 年 10 月 20 日	更新は以下のとおり • 文書内で用いられるすべての画像の更新. 変更は用いられる色や画像のルック・アンド・フィールの一層の共通化を含む • meta-data を metadata に変更. ハイフンを除去. • 図 3.2.a を物理ファイルからコンテンツに更新 • (サブ) マニフェストに関する説明と要件を除去. 新しい説明は, IMS が IMS コンテンツパッケージの新しいバージョンの作業を完了するまで, ADL が (サブ) マニフェストの利用を推奨しないことを含む. 文書全体を通じて ADL Notes を追加. • コンテンツパッケージに含まれるすべてのファイルがマニフェストファイルで宣言ないし参照されなくてはならないという IMS コンテンツパッケージに関する推奨の説明を追加. • <adlcp:dataFromLMS>の SPM を 4096 から 4000 に更新 • adlcp:persistState 要素に関するすべての説明と要件を削除 • href 属性の SPM はそれに適用されるすべての xml base を合わせたものであるという説明を追加. • <dependency>要素の identifierref 属性を, その値が

		(サブ) マニフェスト中のリソース識別子は参照できないという記述となるよう更新. • 3.4.1.26<manifest>要素に, IMS コンテンツパッケージの作業が終わるまで (サブ) マニフェストを使わないよう推奨する ADL Note を追加. • 3.4.2 (サブ) マニフェストを削除. 今後の通知があるまで (サブ) マニフェストをサポートおよび使用するすべての説明と要件を削除. • 3.4.3.1XML Base 要素の扱いを, href と xml base の SPM 定義に関する特別な注意の説明を含むよう更新. • 3.4.3.3 パラメータ要素の扱いを, 正規パラメータ構成のアルゴリズムの修正を含むよう更新. • 表 3.5.3a を, <adlcp:persistState>要素を除くよう更新. • 表 3.5.3a を, 項目 1.4 <metadata>, 1.4.1 <schema>, 1.4.2 <schemaversion>に対する要求条件を O (任意) から M (必須) に変更するよう更新. この表の値は以前定義されていたものと一致しない. • 3.6.2 再利用のための学習コンテンツのパッケージングを削除. この節は目的を達するために (サブ) マニフェストの利用を扱っていた. この節は (サブ) マニフェストに関する更新による他の変更に基づいて削除された. • 4.2.1<lom>要素を, 同じ SCORM コンテンツモデル要素を記述するために複数メタデータインスタンスをどのように保持するかを記述するよう更新. • <language> 要素 (LOM.General.Language)が, コンテンツが言語的なコンテンツでない場合, none の値を持てることを明記. • VCard シンタックスを使うメタデータのコード例示を更新. 要求される&#13 と &#10 のために\n を削除. • 4.2.4.2 <contribute>要素を SPM 値を修正するよう更新. IEEE XML LOM バインディングと合致するよう 30 から 10 に変更. • 4.2.4.2 .2 <entity>要素を SPM 値を修正するよう更新. IEEE XML LOM バインディングと合致するよう 40 から 10 に変更. • 4.2.4.4 <language>要素を, language 要素で保持する値の書式をより明確化するよう更新. • 4.2.8.2 <resource>要素を, IEEE XML LOM バインディングと合致するよう SPM を 0 以上から 0 ないし 1 に変更するよう更新. • 4.2.8.2.2 <description>要素を, IEEE XML LOM バインディングと合致するよう SPM を 0 ないし 1 から 0 以上 (SPM: 10) に変更するよう更新. • 5.1 シーケンシングとプレゼンテーションを,
--	--	--

		(サブ) マニフェストを参照する<item>要素に置かれたシーケンシング情報を扱った説明を除去するよう更新。これは(サブ) マニフェストの更新に関する他の変更により削除された。 • 5.1.7.1.1 <minNormalizedMeasure>要素を, LMS がこの値を cmi.scaled_passing_score データモデル要素の初期化に使わなくてはならないことを記述するよう更新。 • 5.1.11 <rollupConsiderations>要素を, measureSatisfactionIfActive 属性のデフォルト値を偽から真に変更するよう更新。 • 5.2.1 <presentation>要素を, SCO からリソースに変更するように更新。これはリソースが, アセット・SCO 両方に対応するためである。 • <hideLMSUI>要素値空間で用いられるトークンの説明を更新。 • 5.2.1.1.1 <hideLMSUI>要素を新しいトークン値 (suspendAll, abandonAll および exitAll) を含めるように更新。 • このバージョンを SCORM 2004 第3版とするよう文書のバージョンを更新。また, この文書の内部バージョンをバージョン 1.0 に更新。 • この文書を通して必要な個所で SCORM 2004 第3版への変更に関連する情報を更新。 • ADL Co-Lab ハブへのコンタクト情報を更新 • 文書全体を通じた編集的な更新 (例えば SCORM 登録商標の一貫した適用) • 他の SCORM 文書における変更を反映した一般的な更新 • adlnet.org の参照の adlnet.gov の参照への更新。 • SCORM コンテンツモデル要素の5つの要素を5つの節に分離するよう更新。いくつかはひとつにまとめられた。5つの要素と対応する節はアセット, SCO, アクティビティ, コンテンツオーガニゼーション, コンテンツアグリゲーション, である。 • SCORM コンテンツモデル要素の分かれた節への分離に合わせて図 2.1.3a と 2.1.4a を追加。 • <manifest>, <organization>, <item>, <resource>および <sequencing>要素の識別子属性の説明をこれらの識別子がマニフェストの範囲内で大文字小文字によらず (識別子は大文字小文字に依存しない) 一意でなくてはならないことを定義するよう更新。 • <resource>要素の type 属性の使用法に関する要求を変更。<resource>要素を参照する<item>要素の type 属性の値が webcontent に設定されるよう要求を変更。この変更の前は規格準拠のために type 属性は webcontent に設定されなくてはならなかつ
--	--	--

		た。この要求は一般的な実践とされ、SCORM 準拠のためには不要となった。 • <resource>要素の href 属性に URL が追加パラメータを含むことができることを示す説明を追加。 • <file>要素の記述を明確化のために更新。 • 3.4.3.3 節の題名を、URL の一部としての追加パラメータの扱い、に変更。 • SCORM メタデータに関するすべての要求を削除。SCORM はメタデータ定義、SCORM コンテンツモデル要素、マニフェストで定義された SCORM コンテンツモデル要素にメタデータに関連付ける方法の記述のために、まだ IEEE LOM の使用を推奨する。しかし、SCORM は現在、2006/6/16 の米国国防省指示文書 1322.26 遠隔教育の開発、管理、配信で定義された国防省の要求に関連したメタデータ要求に関して ADL レジストリを参照する。ADL はまた、他の実践コミュニティも特定の実践コミュニティで定義されたメタデータ要求を実装するために ADL レジストリを参照することを奨励する。 • コンテンツパッケージマニフェスト中の SCORM コンテンツモデル要素を記述するメタデータインスタンスで用いるときに、必須であった LOM 要素に関して、この要求は、現在、任意に緩和された。必須要求は存在しない。米国国防省指示文書 1322.26 に準拠する必要がある場合、すべての要求は ADL レジストリメタデータ要求条件で定義される。 • SCORM ベストプラクティスと制限ボキャブラリに関する記載を他の SCORM に関する一般的なメタデータのベストプラクティスとともにすべて削除。 • XSD が影響を受ける要素・属性を規定していないシーケンシング要素 (referencedObjective, measureThreshold, condition, attemptLimit, attemptAbsoluteDurationLimit および selectCount) のデフォルト値に関する記載をすべて削除。 • コード例 5-24 を<adlseq:rollupConsiderations>要素を適切な場所に入れるよう更新。 • コンテンツパッケージの<schemaversion>要素の値に対する要求を変更。現在、値は 2004 3rd Edition でなくてはならない。この値は、コンテンツパッケージが準拠する SCORM のバージョンを表す。 • <file>要素に、コンテンツパッケージ中のファイルについては、<file>要素の href 属性は、<resource>要素の href 属性からそれに含まれるパラメータを除いたものに等しいものにしなければならないという説明を追加。 • href の値で大文字・小文字を区別するシステムが
--	--	--

		あるので、開発者やツール作成者は href の値を定義する際注意するよう説明を追加. • <ruleCondition>要素の繰り返し数に関して定義された要求を IMS が提供した XSD に対応付けるよう説明を更新. <ruleCondition>要素は <ruleConditions>要素中で 1 回以上定義されなくてはならない. 要求条件は 0 回以上となっていた. もし, <preConditionRule>, <postConditionRule>または <exitConditionRule>が提供される場合, 条件も提供されなくてはならない. • referencedObjective 属性の説明を, 値が提供されていればこの値は, アクティビティに対して定義された<primaryObjective> か<objective>要素のひとつのいずれかの objectiveID に一致しなくてはならないことを示すように更新. • <ruleCondition>要素の繰り返し数に関して定義された要求を IMS が提供した XSD に対応付けるよう説明を更新. <ruleCondition>要素は <ruleConditions>要素中で 1 回以上定義されなくてはならない. 要求条件は 0 回以上となっていた. もし, <preConditionRule>, <postConditionRule>または <exitConditionRule>が提供される場合, 条件も提供されなくてはならない. • IETF RFC 2396 への参照を IETF RFC 3986 (より新しいバージョン) 変更.
--	--	---

このページは空白である。