



ADL SCORM Version 1.3
アプリケーションプロファイル

ワーキングドラフト 1.0
2003年3月26日

日本語版ドラフト 0.94
2003年7月15日

このページは空白である。

DRAFT

Advanced Distributed Learning
ADL SCORM Version 1.3
アプリケーションプロファイル

www.adlnet.orgから入手可能

日本語版はwww.alic.gr.jp, www.elc.or.jp から
入手可能

質問やコメントは ADLNet 問い合わせセンターまで。
日本語版に関する質問やコメントは ALIC, eLC まで。

このページは空白である。

DRAFT

日本語版翻訳者

ALIC 相互運用性部会 eLC 相互運用性委員会

氏名	所属	担当
池田 満	北陸先端科学技術大学院大学	7章
石打 智美	NTT データ (株)	7章
関口 法門	(株) 富士ゼロックス総合教育研究所	1~3章
高橋 和彦	(株) 富士通インフォソフトテクノロジー	8~9章
樋田 稔	エスエイティーティー (株)	査読
仲林 清	(株) エヌ・ティ・ティ エックス	7章
林 雄介	北陸先端科学技術大学院大学	7章
星野 忠明	エスエイティーティー (株)	6章
平田 謙次	産業能率大学	4章
廣澤 晃一	(株) マイクロメイツ	査読
廣瀬 文男	ウイルソンラーニングワールドワイド (株)	5章

DRAFT

このページは空白である。

DRAFT

目次

目次	V
摘要	VII
1章 概要	1-1
1.1. 概要	1-3
1.1.1. 参加方法	1-4
2章 SCORM コンテンツアグリゲーションモデル	2-1
2.1. コンテンツアグリゲーションモデル	2-3
2.1.1. コンテンツモデル概要	2-3
2.1.2. SCORM メタデータ構成要素	2-9
3章 メタデータ	3-1
3.1. メタデータ	3-3
3.1.1. 概要	3-3
3.1.2. 1484.12.1-2002 情報モデル	3-4
3.1.3. SCORM メタデータ XML バインディング	3-22
3.1.4. SCORM メタデータアプリケーションプロファイル	3-87
4章 シーケンシングの概要	4-1
4.1. 概要	4-3
4.2. シーケンシング用語	4-4
4.2.1. 学習アクティビティ / アクティビティ	4-4
4.2.2. アクティビティのトラッキング	4-5
4.2.3. 能動的 / 受動的コンテンツ	4-6
4.2.4. アクティビティツリー	4-6
4.2.5. クラスタ	4-8
4.2.6. アクティビティの経過のトラッキング	4-9
4.2.7. シーケンシング定義モデル	4-11
4.2.8. コンテンツタイプ	4-11
5章 シーケンシング定義モデル	5-1
5.1. SCORM シーケンシング定義モデル	5-3
5.1.1. シーケンシングコントロールモード	5-3
5.1.2. シーケンシングルール	5-8
5.1.3. 制限条件	5-13
5.1.4. 補助リソース	5-18
5.1.5. ロールアップルールについて	5-19
5.1.6. ロールアップコントロール	5-24
5.1.7. 学習目標	5-25
5.1.8. アクティビティの選択の制御	5-32
5.1.9. ランダム化コントロール	5-33
5.1.10. 配信コントロール	5-35
6章 シーケンシング XML バインディング	6-1
6.1. XML バインディングシーケンシング定義モデル	6-3
6.1.1. <sequencing> 要素	6-3
6.1.2. <sequencingCollection> 要素	6-4
6.1.3. コンテンツパッケージとの関係	6-5

7章 シーケンシング動作.....	7-1
7.1. シーケンシング動作.....	7-3
7.1.1. トラッキングモデル.....	7-4
7.1.2. オーバーオールシーケンシングプロセス.....	7-17
7.1.3. ナビゲーション動作.....	7-20
7.1.4. 終了動作.....	7-24
7.1.5. ロールアップ動作.....	7-28
7.1.6. 選択ランダム化動作.....	7-36
7.1.7. シーケンシング動作.....	7-38
7.1.8. 配信動作.....	7-44
8章 ナビゲーションユーザインタフェース管理.....	8-1
8.1. ユーザインタフェース管理の背景.....	8-3
8.2. ナビゲーションの背景.....	8-3
8.2.1. ナビゲーションの概観.....	8-4
8.2.2. ナビゲーションイベント.....	8-5
8.3. 補助サービスの背景.....	8-8
8.3.1. 補助サービスの概観.....	8-8
8.4. ユーザインタフェース装置の管理.....	8-9
8.4.1. ナビゲーションのためのユーザインタフェース装置の管理.....	8-9
8.4.2. 補助サービス用のユーザインタフェース装置の管理.....	8-10
8.4.3. ユーザインタフェース状態またはイベントの交信.....	8-13
8.4.4. ナビゲーションによる学習リソースの終了.....	8-18
9章 ランタイム環境データモデル.....	9-1
9.1.1. 序説.....	9-3
9.1.2. SCORM ランタイム環境データモデル.....	9-3
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 改定履歴.....	C-1
改定履歴.....	C-3

摘要

このドキュメントの内容は、ドラフトと考えられるものである。このドキュメントは、さまざまな標準や仕様についてのワーキンググループで進行中の作業に基づいて変更されることがあるものとする。

このドキュメントは、SCORM バージョン 1.3 アプリケーションプロファイルを定義するものである。それは本来技術的で、コンテンツ作成者や学習デザイナー、学習管理システム (LMS) ベンダー向けにできている。このドキュメントは、IMS シンプルシーケンシング仕様 (以下 IMS SS 仕様と表記) の統合、IMS SS 仕様の統合をサポートする SCORM ランタイム環境データモデルの変更、および全体的な整理と意味の明確化、が行われている。このドキュメントは SCORM バージョン 1.3 に先駆けて供給される。

このドキュメントは SCORM バージョン 1.3 のワーキングドラフトである。このドキュメントは、最終の SCORM バージョン 1.3 作成プロセス (2003 年終り頃に予定されている) を始めるものである。このドキュメント (あるいはその形式) は、今後数ヶ月間でさらに発展するだろう。このドラフトの発展を成功させることは、以下の項目に委ねられている。

1. *ADL コミュニティーによって実行されたレビューとプロトタイプ*。ADL は、様々なステークホルダーがシーケンシングエンジンやシーケンシングルールに沿ったコンテンツを開発することを期待している。ADL 技術チームは、ドラフトに関する様々な問題を引き出すために、プロトタイプを使用し評価する。これらの問題およびコメントは、最終の SCORM バージョン 1.3 で評価され修正される。
2. *新しい基準と仕様が安定したときそれを包含すること*。これは、IMS コンテンツパッケージングのメンテナンス努力、コンテンツオブジェクト通信のためのデータモデルである IEEE 1484.11.1 ラーニングテクノロジードラフト基準、コンテンツとランタイムサービスの通信のための ECMA スクリプト API の IEEE 1484.11.2 ドラフト基準、学習オブジェクトメタデータ (Learning Object Metadata: LOM) データモデルのための拡張可能マークアップ言語 (eXtensible Markup Language: XML) バインディングの IEEE 1484.12.3 ドラフト基準を含んでいる。

現在、最終の SCORM バージョン 1.3 および準拠テストスイートがリリースされるまで、確実な準拠基準は開発できず、準拠の主張やテストの検証もできない。この過程で、ADL コミュニティーが SCORM バージョン 1.3 の発展をテストするために、テストスイートのベータリリースが利用可能となるだろう。

このページは空白である。

DRAFT

1 章 概要

DRAFT

このページは空白である。

DRAFT

1.1. 概要

SCORM バージョン 1.3 アプリケーションプロファイルは、SCORM バージョン 1.3 の最終バージョンに含まれる前に提案されている変更点について、テストと評価を実行するために ADL コミュニティーに提供されている。

このドキュメントの目的は、SCORM バージョン 1.3 へ変更や追加を紹介することである。追加点および変更点は以下の通りである。

SCORM コンテンツアグリゲーションモデル

- コンテンツモデル
 - SCA (Sharable Content Asset, 共有可能コンテンツアセット) という新しい概念
 - 新しい SCORM メタデータ構成要素の追加(SCA, アクティビティ)
 - 全体的な整理と意味の明確化
- メタデータ
 - IEEE 1484.12.1-2002 学習オブジェクトメタデータ (LOM: Learning Object Metadata) のサポートのための変更の統合
 - SCORM メタデータ情報モデルの更新
 - SCORM メタデータ XML バインディングの更新
 - SCORM メタデータアプリケーションプロファイルの更新(SCA と アクティビティ)
 - 全体的な整理と意味の明確化
- コンテンツパッケージング
 - 起動可能な学習資源についての用語の変更(Asset – SCA)
 - 葉ノードのみが学習資源への参照を許可するように変更(IMS SS 仕様を基とする)
 - アイテム属性のパラメータの使用法の明確化(*To be supplied*)
 - xml:base 要素の使用法の明確化(*To be supplied*)
 - 全体的な整理と意味の明確化
- シーケンシング
 - IMS SS 仕様の統合
 - IMS SS 仕様を補う基本的なナビゲーション機能の導入

SCORM ランタイム環境 (RTE)

- IMS SS 仕様を補足する SCORM ランタイム環境データモデルの変更
- 全ての SCORM ランタイム環境データモデル要素の必須な実装に必要とされる変更
- 全体的な整理と意味の明確化

このアプリケーションプロファイルは、SCORM コンテンツパッケージングを使用してシステム間で交換することができる、順序付けされた学習コンテンツの開発において、ADL コミュニティーをガイドするために使われるものである。このドキュメントが定義するシーケンシングアプリケーションプロファイルが、SCORM バージョン 1.3 に組み込まれるであろうことが予想される。

この SCORM バージョン 1.3 アプリケーションプロファイルはドラフトバージョンである。このドキュメントは、まだ進行中の作業と考えられるべきである。IMS ワーキンググループや IEEE ワーキンググループの作業で、このドキュメントのコメント期間中に変更が生じるかもしれない。このような変更は、もしあれば、適切に SCORM1.3 アプリケーションプロファイルに補足のリリースとして反映されるだろう。ADL は、ADLNet.org のフォーラムエリアにある SCORM1.3 アプリケーションプロファイルのウェブ掲示板への投稿によって、ADL の技術チームにこのアプリケーションプロファイルまたはサンプルについてのフィードバックが提出されることを求めている。

IMS SS 仕様は、現在“Final Public Release”の状態である。IMS 仕様は“Public Draft”の状態のときから、ADL は仕様を評価しテストするために 3 つの同時並行的な活動に従事している。

- シーケンシングエンジンのトライアル版開発と、SCORM バージョン 1.3(ベータ版)サンプルランタイム環境の統合。
- シーケンシング動作を図解する多くのコンテンツの例を開発した。これらのコンテンツの例は、SCORM バージョン 1.3(ベータ版)サンプルランタイム環境でテストされ実証された。
- SCORM バージョン 1.3 アプリケーションプロファイル。このドキュメントを執筆することは、IMS SS ワーキンググループへの ADL の参加によって通知された。

1.1.1. 参加方法

ADL 技術チームは、以下の方法で SCORM バージョン 1.3 アプリケーションプロファイルをさらに開発し定義することに参加してくれるようにベンダーに依頼している。

- **SCORM バージョン 1.3 アプリケーションプロファイルの実装:** シーケンシングをサポートする LMS とコンテンツを作成する時、IMS SS 仕様に加えて SCORM バージョン 1.3 アプリケーションプロファイルを使用すること。
- **コンテンツの実際例へのアプリケーションプロファイルの適用:** IMS SS ワーキンググループは、古典的な行動トレーニング、発見学習および協調的な問題解決のような様々な教育学的、教授法的アプローチで見つかった、シーケンシング動作に広く用いられているセットに注目した。ADL 技術チームは、コンテンツを作成する時 IMS SS 仕様と結合したこの

アプリケーションプロファイルを適用し，かつ SCORM バージョン 1.3 アプリケーションプロファイルがサポートしないユースケースを提出するよう，コンテンツベンダーに依頼している．これらのユースケースは，将来の開発課題として IMS にフィードバックされるだろう．

ADL フォーラムでのフィードバックの供給: ADL 技術チームは，SCORM バージョン 1.3 アプリケーションプロファイルのドキュメントおよびアプリケーションプロファイルの実装で得た知識や経験に関する，どんなフィードバックおよび（または）質問を求めている．

DRAFT

2 章

SCORM コンテンツアグリゲーション モデル

DRAFT

このページは空白である。

DRAFT

2.1. 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、教育設計者や実装者が学習資源を集約して適切な学習体験を提供するための、教育手法に依存しない手段である。ここで、学習資源とは学習体験のサポートで利用される情報のあらゆる表現である。学習体験は電子的ないし非電子的な学習資源によって支援されるアクティビティによって成り立っている。

学習資源を生成・提供するプロセスとして、単純な電子的アセットを生成、検索し、これらをさらに複雑な学習資源にまとめあげ、あるいは集約し、学習資源を決められた順序で配布する過程がある。SCORM バージョン 1.2 のコンテンツアグリゲーションモデルは、しっかりしたシーケンシングモデルがサポートされていなかった。IMS SS 仕様を追加することで、SCORM バージョン 1.3 ではコンテンツアグリゲーションモデルにこの重要な要素が加わる。

SCORM バージョン 1.2 コンテンツアグリゲーションモデルは、以下の内容を含んでいる。

- コンテンツモデル: コンテンツアグリゲーションにおける構成要素を規定する用語
- メタデータ: コンテンツアグリゲーション構成要素の特定のインスタンスの記述を行うための仕組み
- コンテンツパッケージング: 意図したコンテンツアグリゲーションの記述方法（コンテンツ構造）、および異なる環境に移植するために学習資源をパッケージする方法（コンテンツパッケージング）。

SCORM バージョン 1.3 コンテンツアグリゲーションモデルは、バージョン 1.2 コンテンツアグリゲーションモデルに以下の変更を組み込んでいる。

- IMS SS 仕様に準拠した用語の追加
- シーケンシング定義モデルの追加。これは、アクティビティをどう並べて配信するかを定義するルールベースのモデルである。このモデルは、実行時に適用され、結果として配信アクティビティを特定する、動作の集合を定義する。特定されたアクティビティに対して、それに関連付けられたリソースが起動される。
- IEEE 1484.12.1-2002 学習オブジェクトメタデータ (LOM) 標準に基づいたメタデータの変更

2.1.1. コンテンツモデル概要

SCORM コンテンツモデルは、再利用可能な学習資源から学習体験を構成するために使用される SCORM の構成要素である。また、コンテンツモデルはこれらの共有可能かつ再利用可能な学習資源の要素を集約して、高レベルの学習ユニ

ットを作成する方法を規定する。コンテンツモデルの構成要素はすべて学習資源の特殊化したものとみなされる。SCORMバージョン 1.3 アプリケーションプロファイルは、以下の構成要素からなる: アセット, SCA (Sharable Content Asset, 共有可能コンテンツアセット), SCO (Sharable Content Objects, 共有可能コンテンツオブジェクト), コンテンツアグリゲーション。

2.1.1.1 アセット

学習資源の最も基本的なものはアセットである。アセットは、Web クライアントで表現されるテキスト、画像、音声、アセスメントオブジェクト、の電子的な表現やデータの断片である。SCORM 準拠の LMS では、個々のアセットの起動は行われない。

アセットの属性や特徴は、アセットメタデータ (以下のアセットメタデータ参照) で記述され、リポジトリの中での検索・発見が可能となり、これによって再利用が可能になる。アセットをアセットメタデータに結びつける仕組みは、SCORM で記述されているコンテンツパッケージである。

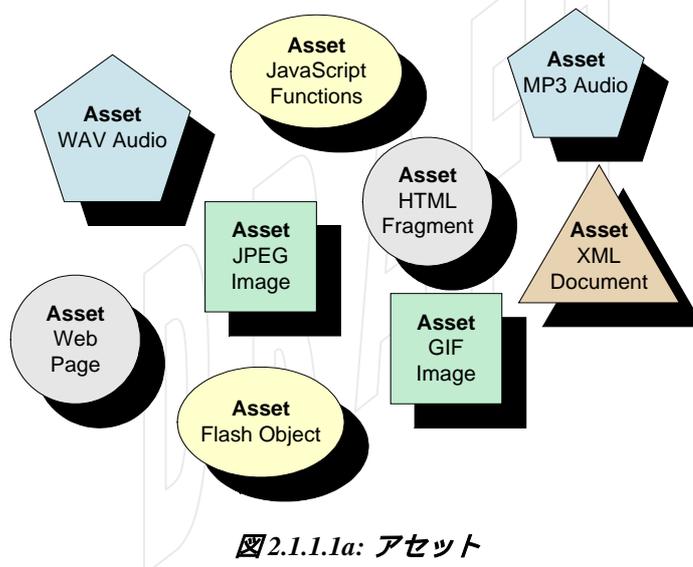


図2.1.1.1a: アセット

2.1.1.2 SCA (Sharable Content Asset, 共有可能コンテンツアセット)

SCA は SCORM バージョン 1.3 から導入された新しい用語であるが、概念は決して新しいものではない。これまで、LMS と通信しない学習資源のことをどのように呼ぶかで少し混乱があった。SCORM バージョン 1.2 では、これはアセットと呼ばれていた。しかしながらアセットという用語は、Web クライアントで表現されるテキスト、画像、音声、アセスメントオブジェクト、の電子的な表現やデータの断片であると定義されている。

それ故に、SCA はひとつあるいは複数のアセットで構成されるひとつの起動可能な資源の集合を定義する。以下の図 2.1.1.2a は、いくつかのアセットで構成された SCA の例を示している。SCO (Sharable Content Object, 共有可能コンテン

ツオブジェクト)との違いは、SCAはSCORM通信APIアダプタでLMSと通信をしないということである。

再利用性を向上するために、SCAは学習文脈から独立してはいなくてはならない。例えばSCAは異なる学習資源において異なる学習目標のために使用できる。さらに、アクティビティ(コンテンツアグリゲーションを参照)は1つ以上のSCA(および/またはSCO)を集約してより高いレベルの学習訓練ユニットを構成し、より高いレベルの学習目標を満たすことができる。

複数の学習目標に再利用できるように、SCAは主観的に小さなユニットとなるように意図されている。SCORMはSCAの正確な大きさについて特定の制約を課さない。コンテンツ設計、作成の過程を通してSCOの大きさを決定する際、ある学習目標を満たすことに要するコンテンツの最小の論理的な大きさを検討する必要がある。コンテンツ開発者は、SCAの大きさを、学習成果をあげるためにどれだけの情報が必要とされるか、どの程度の再利用性が求められるか、によって決める必要がある。

SCAの属性や特徴はSCAメタデータで記述され、リポジトリの中での検索・発見が可能となり、これによって再利用が可能になる。SCAをSCAメタデータに結びつける仕組みは、SCORMで記述されているコンテンツパッケージである。

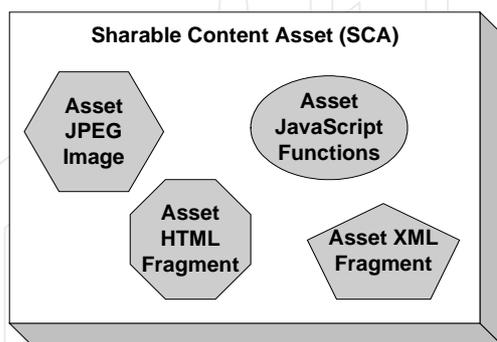


図2.1.1.2a: Sharable Content Asset (SCA)

2.1.1.3 SCO (Sharable Content Object, 共有可能コンテンツオブジェクト)

SCOはSCORMランタイム環境を利用してLMSと通信を行うひとつの起動可能な学習資源を含む、ひとつもしくは複数のアセットの集合である。SCOは、SCORMランタイム環境データモデルによってLMSが動作を記録できる最小単位の学習資源である。SCOとSCAの違いは、SCOはSCORMランタイム環境の通信APIを使用してLMSと通信するという点である。以下の図2.1.1.3aは、複数のアセットで構成されるSCOの例を示している。

再利用性を向上するために、SCOは学習文脈から独立してはいなくてはならない。例えばSCOは異なる学習資源において異なる学習目標のために使用できる。さらに、アクティビティ(コンテンツアグリゲーションを参照)は1つ以上の

SCO（および/または SCA）を集約してより高いレベルの学習訓練ユニットを構成し、より高いレベルの学習目標を満たすことができる。

複数の学習目標に再利用できるように、SCO は主観的に小さなユニットとなるように意図されている。SCORM は SCO の正確な大きさについて特定の制約を課さない。コンテンツ設計、作成の過程を通して SCO の大きさを決定する際、実行時に LMS によって動作を記録したいコンテンツの最小の論理的な大きさを検討する必要がある。コンテンツ開発者は、SCO の大きさを、学習成果をあげるためにどれだけの情報が必要とされるか、どの程度の再利用性が求められるか、によって決める必要がある。

SCO の属性や特徴は SCO メタデータ（以下の SCO メタデータ参照）で記述され、リポジトリの中での検索・発見が可能となり、これによって再利用が可能になる。SCO を SCO メタデータに結びつける仕組みは、SCORM で記述されているコンテンツパッケージである。

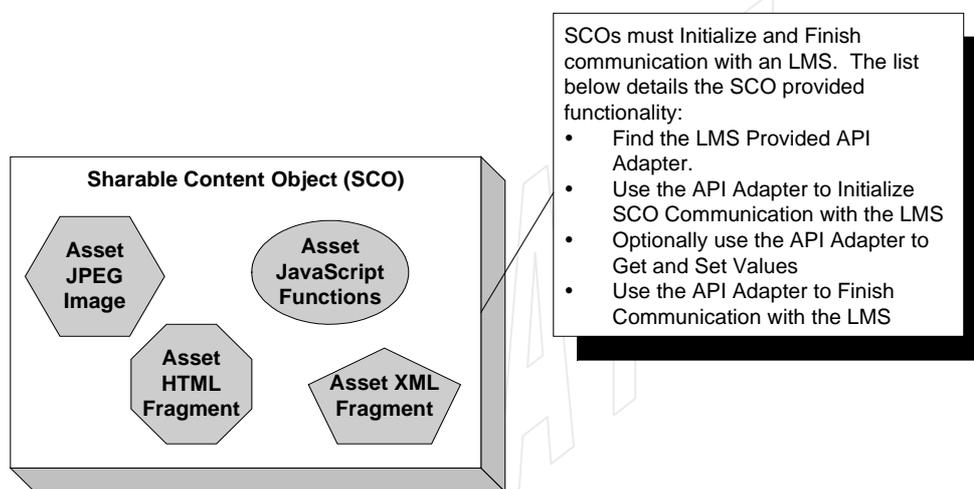


図 2.1.1.3a: Sharable Content Object (SCO)

SCO は SCORM ランタイム環境を遵守する必要がある。これは SCO が LMS の通信 API アダプタを見つけて、最小限の API 呼び出し (LMSInitialize(“”)と LMSFinish(“”)) を行わなくてはならないことを意味する。それ以外の API 呼び出しはオプションで、コンテンツの特性に依存するため、実装は必須ではない。

SCO が SCORM ランタイム環境で動作するための要求条件は以下の利点をもたらす。

- SCORM ランタイム環境をサポートするすべての LMS は、誰が作成した SCO であれそれを起動し、動作を記録することができる。
- SCORM ランタイム環境をサポートするすべての LMS は、SCO の動作を記録し、それがいつ起動し、いつ終了したかを知ることができる。
- SCORM ランタイム環境をサポートするすべての LMS は、どの SCO も同様の方法で起動できる

詳細は、SCORMバージョン 1.2 のランタイム環境の章を参照のこと。

2.1.1.4 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、学習ユニット（アクティビティ）の結合の記述、他のアクティビティとの関係、および学習分類を集約したアクティビティ（すなわち、コース、章、モジュールなど）のマップ（コンテンツ構造）である。図 2.1.1.4a にコンテンツアグリゲーションの例を示す。

他のアクティビティを含んでいるアクティビティから成るコンテンツアグリゲーションは、任意の深さで入れ子になる。他のアクティビティを含むアクティビティはクラスタとよばれる。クラスタは、アクティビティを結合して学習ユニットとしてグループ化するが、学習資源を関連づけてはいない。他のアクティビティを含まないアクティビティ（ツリーの葉の位置にあるアクティビティ）は、学習資源（SCA または SCO）との関連をもつだろう。

コンテンツアグリゲーションの属性や特徴は、コンテンツアグリゲーションメタデータで記述される。このメタデータを用いてリポジトリの検索・発見が可能となり、コンテンツアグリゲーションの再利用が可能になる。コンテンツアグリゲーションをコンテンツアグリゲーションメタデータに結びつける仕組みは、SCORM で記述されているコンテンツパッケージである。

コンテンツアグリゲーションの中のそれぞれのアクティビティの属性や特徴は、アクティビティメタデータで記述される。このメタデータによってリポジトリの中での検索・発見が可能となり、アクティビティの再利用が可能になる。アクティビティをアクティビティメタデータに結びつける仕組みは、SCORM で記述されているコンテンツパッケージである。

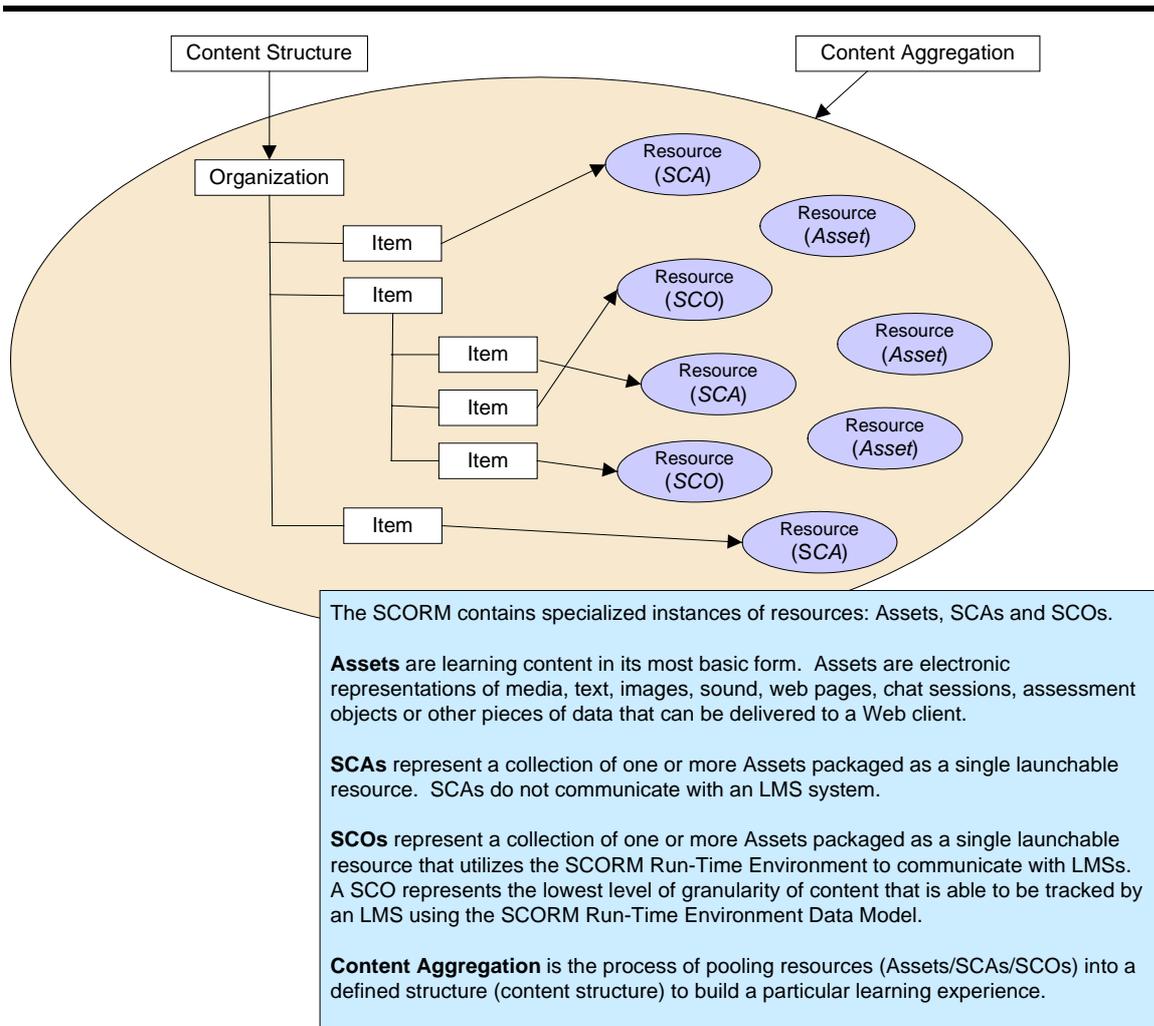


図2.1.1.4a: Content Aggregation

シーケンシングは、アクティビティだけに適用される。アクティビティの意図したシーケンシングは、他のアクティビティとの関係の中の構造化されたアクティビティによって、また、それぞれのアクティビティのシーケンシング情報を結合して、コンテンツアグリゲーションの一部として定義される。LMSは、コンテンツアグリゲーションに記述されたシーケンシング情報を解釈すること、および実行時の学習資源（学習経験）の現在のシーケンシング（実際の順序付け）をコントロールするためのシーケンシング動作（第4章「シーケンシング概要」を参照）を適用する責任を負う。

これは、教材をスタンドアロン CBT（Computer-Based Training）作成ツールを利用して開発する手法からの大きな飛躍を意味する。過去においては、これらのツールは一般的に、学習者が次に教材のどの部分を見るかを制御するすべてのナビゲーション情報を、専用のデータフォーマットに埋め込んでいた。ほとんどすべての場合、教材作成ツールやシステムは、専用でしばしば特殊なコースシーケンシング手法を定義・実装していた。これによって、異なる教材作成環境間でコンテンツを共有することは、現在でも非常に難しいか、まったく不可

能である。また同様に、コンテンツを異なるシーケンシング要求を有する文脈で再利用することも困難である。

ブラウザベースの SCORM では、シーケンシング情報はコンテンツアグリゲーションで表現されるアクティビティで定義され、アクティビティから成る学習資源の外側にある。定義されたシーケンシング動作を適用し、アクティビティから成る学習資源を起動するのは LMS である。この概念は重要である。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資源に埋め込まれていたら、学習資源の再利用は不可能だからである。例えば、学習資源に他の学習資源への分岐情報が含まれていたら、この学習資源は分岐先の学習資源が存在しない他のコースでは利用できない。学習資源が再利用可能であるためには、それが独立して自己完結している必要がある。

しかし SCORM では、ある種の学習資源は特定の学習タスクを実現する内部ロジックを含むことができる。そのようなコンテンツは学習者の反応に応じて、コンテンツ内部で分岐することができる。このような分岐はすべて学習資源内部で閉じているため、学習資源が独立（従って再利用可能）であることと整合しており、通常 LMS からは感知できない。このような SCO 内部での分岐は、（その学習資源が）他のコンテンツアグリゲーションに存在するしないを問わず、外部の学習資源を参照してはならない。これは、どのような学習資源を使い、それをどのように集約するかを決定する際に、コンテンツ開発者が注意を払うべき重要なポイントである。

2.1.2. SCORM メタデータ構成要素

メタデータは、SCORM コンテンツモデルの各要素と IEEE LTSC 学習オブジェクトメタデータ規格の項目を対応付け、望ましい使用法を示したものである。この対応付けの詳細は SCORM で説明されている。一般的に、アセット、SCA、SCO、学習アクティビティ、コンテンツアグリゲーションをシステム内およびシステム間で検索・発見して共有・再利用するために、これらにメタデータを付与して属性や特徴の一貫性のある記述を行うためのガイドが提供されている。

再利用を意図している組織は、組織内でメタデータをどのように付与するかの方針を規定しなくてはならない。SCORM はコンテンツモデル構成要素のメタデータタグ付けの方針に関する要求条件を強要するものではなく、共有・再利用を図ろうとする組織に標準に基づく実際的なガイドを提供することをねらいとしている。

2.1.2.1 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を記述する。コンテンツアグリゲーションメタデータを付与する目的は、例えば教材リポジトリの中でコンテンツアグリゲーションを発見可能とし、コンテンツ

アグリゲーションで定義されるコンテンツ構造の属性や特徴を記述するための情報を提供することである。

2.1.2.2 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、コンテンツアグリゲーション内でそれぞれ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付与する目的は、教材リポジトリの中でアクティビティをアクセス可能（すなわち発見可能）とすることである。メタデータは、アクティビティを全体として記述すべきである。これは、SCORMバージョン 1.3 で追加された新しいメタデータのアプリケーションプロファイルである。アクティビティのために構築された任意のメタデータの必要条件は、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で述べられた必要条件と同一でなくてはならない。

注記: アクティビティは SCORM バージョン 1.3 で導入された新しい用語である。

2.1.2.3 SCO メタデータ

SCO に適用されるメタデータの定義で、SCO で表現されるコンテンツの属性や特徴の記述情報を提供する。例えば、コンテンツリポジトリからこのようなコンテンツを発見して再利用することを容易にする。

2.1.2.4 SCA メタデータ

SCA に適用されるメタデータの定義で、SCA で表現されるコンテンツの属性や特徴の記述情報を提供する。例えば、コンテンツリポジトリからこのようなコンテンツを発見して再利用することを容易にする。

注記: SCA は SCORM バージョン 1.3 で導入された新しい用語である。これについてコンテキストに依存しない方法で記述する場合、SCORM バージョン 1.2 で議論された SCO メタデータプロファイルが使用されるべきである。

2.1.2.5 アセットメタデータ

加工していないメディアアセットに適用されるメタデータの定義で、アセットの教材内での利用法には依存しない属性や特徴の記述情報を提供する。このメタデータは、例えば教材作成時にコンテンツリポジトリからこれらのアセットを発見して再利用することを容易にする。

2.1.2.6 メタデータの使用方法

上に述べたコンテンツモデル構成要素を対応するメタデータアプリケーションプロファイルに結びつける仕組みは、SCORM で記述されているコンテンツパッケージである。それらは現在コンテンツパッケージの中に 5 箇所ある。

-
- **マニフェスト:** マニフェストレベルのメタデータは IEEE LTSC LOM バインディングに準拠しなければならないが, SCORM による追加制限は無い. このメタデータは SCORM の範囲外であり, 前述のカテゴリーに入らない.
 - **オーガニゼーション:** オーガニゼーションレベルのメタデータは, 全体としてコンテンツアグリゲーションを記述する. これは, コース, ユニット, レッスンあるいは他の系統立てた教育ユニットでありうる. オーガニゼーションレベルに置かれるメタデータは, SCORM コンテンツアグリゲーションメタデータである.
 - **アイテム:** アイテムレベルのメタデータは, アクティビティの入れ子階層をコンテキストに依存した方法で記述する. アイテムに関連付ける場合, SCORM アクティビティメタデータ定義を使わなければならない.
 - **リソース:** リソースレベルのメタデータは, コンテキストに依存しない方法で SCO や SCA を記述する. このメタデータは, SCORM SCO メタデータもしくは SCORM SCA メタデータ定義 (アイテムの `adlcp:scormtype` の記述によって決定されたもの) のいずれかにバインドされる.
 - **ファイル:** ファイルレベルのメタデータは, コンテキストに依存しない方法でアセットを記述する. このメタデータは, SCORM アセットメタデータ定義にバインドされる.

DRAFT

3 章 メタデータ

DRAFT

このページは空白である。

DRAFT

3.1. メタデータ

この章では、学習資源へのメタデータの適用に関する詳細なガイダンスを行う。ここで定義される SCORM メタデータアプリケーションプロファイルは、IEEE LTSC 学習オブジェクトメタデータ (LOM) 規格を参照している。IEEE LTSC LOM 仕様は、よく利用される、約 64 のメタデータ要素を定めている – これは多くの応用で実用的な要素数以上である。この章では、SCORM コンテキストにおいて、アセット、SCO、SCA、アクティビティ、学習資源のコンテンツアグリゲーションにタグ付けを行うメタデータとしてどのデータ要素が必須か定義する。また SCORM は、IEEE LTSC LOM 仕様をそのまま取り込むが、SCORM の環境で使用するためのガイダンスもこの章で行う。

この章は 4 つの節からなる。

- 3.1.1 概要 では、LOM の発展の背景について述べる。
- 3.1.2 情報モデル では、メタデータ要素を定義する。これは、各要素の使用法を定義したメタデータタグの「辞書」である。情報モデルは IEEE LTSC LOM 仕様を採用し、これらのタグのエンコード方法は定義しない。
- 3.1.3 SCORM メタデータ XML バインディング は進行中の作業であり、この節には今は情報は含まれていない。この節が完成する時、このドキュメントの最新版が利用できるだろう。
- 3.1.4 SCORM メタデータアプリケーションプロファイル では、SCORM 環境でメタデータを実装するための詳細なガイダンスを行う。ここでは SCORM の必須要素と、それらを SCORM 準拠とするためのエンコード方法を定義する。

3.1.1. 概要

メタデータ (データについてのデータ) の目的は、学習資源の属性や特徴を共通の方法で記述することを可能にする、共通の用語を提供することである。メタデータはカタログ (リスト) にすることができ、それが記述する学習資源と共にパッケージ化できる。メタデータによって記述された学習資源は、利用あるいは再利用のためにシステムティックに検索 / 取得することができる。

3.1.1.1 SCORM メタデータの歴史

学習資源のメタデータは、ここ数年、国内外の多くの組織によって開発されてきた。ADL は、IEEE LTSC Standard for Information Technology -- Education and Training Systems – 学習オブジェクトとメタデータワーキンググループ、the IMS Global Learning Consortium, Inc. (IMS) および Alliance of Remote Instructional

Authoring and Distribution Networks for Europe (ARIADNE) に注目してきた。これらの組織は協力して、本書で取り扱う仕様の核となるセットの開発を行った。

SCORM は、1484.12.1-2002 (IEEE LTSC LOM 標準) を参照している。LOM は、IMS と ARIADNE が学習資源記述のために使用する、メタデータの標準セットを協力して定義する際に開発された。SCORM は、IEEE LTSC LOM 標準と同じメタデータ要素のセットを採用した。SCORM はさらに、バイディング仕様が利用可能になるような時は、そのバイディング仕様を参照するだろう。バイディング仕様は、IEEE LTSC LOM 標準のための XML 表現を供給するだろう。

SCORM は、IMS メタデータ要素の定義を 5 つのコンテンツモデルの構成要素であるアセット、SCO、SCA、アクティビティ、コンテンツアグリゲーションに適用している。これら 5 つの構成要素が、SCORM コンテンツアグリゲーションモデルのメタデータ部分を定義しているのである。

この IEEE から SCORM コンテンツアグリゲーションモデルへの標準化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の間の失われたリンクを提供するものである。次節では、IEEE の定義を SCORM コンテンツアグリゲーションモデルのメタデータ部分に適用した SCORM アプリケーションの定義を行う。

3.1.2. 1484.12.1-2002 情報モデル

1484.12.1-2002 (LOM 標準) 情報モデルは、SCORM 準拠メタデータレコードを形成するために定義されるデータ要素の最初のセットを記述している。情報モデルに定義された必要条件に加えて、SCORM はいくつかのタイプのメタデータインスタンスのためのアプリケーションプロファイルを定義する。これらのアプリケーションプロファイルの必要条件は「3.1.4 SCORM 準拠メタデータレコード」に記載されており、それは「2.2.3.4 XML 拡張メカニズム」で書かれるような追加のデータ要素を含む場合もある。1484.12.1-2002 情報モデルは、9 つのカテゴリに分割される。これらのカテゴリは、1484.12.1-2002 での定義に基づいている。メタデータ要素の 9 つのカテゴリは、以下のようなものである。

1. 一般 (*General*) : 資源を総括して記述する一般的な情報
2. ライフサイクル (*Lifecycle*) : 資源の経歴や現状に関する特徴、資源の開発に関わった人物等に関する情報
3. メタメタデータ (*Meta-metadata*) : (資源ではなく) メタデータそのものの記録に関する情報
4. 技術的事項 (*Technical*) : 資源の技術的要求条件や特徴
5. 教育的事項 (*The Educational*) : 資源の教育的・教育学的特徴
6. 権利 (*Rights*) : 資源の所有権や利用条件
7. 他オブジェクトとの関連 (*Relation*) : この資源と他オブジェクトとの関連付けの定義

8. 注釈 (Annotation) : 教育的仕様におけるコメントとそのコメントの作成者・作成年月日に関する情報
9. 分類体系 (Classification) : 特定の分類体系のどこに属するかについての記述

1484.12.1-2002 情報モデルリストは、メタデータ要素を列挙し、それがどのように階層的に組織化されているかを示している。各要素は、次のような情報について記述されている。

- 項番 (Nr) : 階層項番
- 名前 (Name) : 要素名
- 説明 (Explanation) : 要素の詳細説明
- 繰り返し (Multiplicity) : 親要素の中にいくつの要素のインスタンスが許可されているか
- データ型 (Data Type) : 要素の値が、テキストか数字か日付かといった、サイズやフォーマットの制約。情報モデルでは、言語文字列 (LangString Type) , 日付形式 (Date Type) , 期間形式 (Duration Type) ボキャブラリ (Vocabulary Type) の4つの一般的な型が使用される。これらのデータ型の情報モデルは、後述の SCORM メタデータ情報モデルで指定される。

いくつかの要素には、繰り返し (Multiplicity) とデータ型 (Data Type) のカラムにおいて、“最低限保証すべき最大値 (smallest permitted maximum)” が設定されている。最低限保証すべき最大値は、アプリケーションにおいてメタデータでサポートされているエントリ数に最大数を課すことを意味するが、エントリの最大数は最低限保証すべき最大値の値より大きくなければならない。

ボキャブラリ型のデータ型をもつ要素には、ボキャブラリが制限 (Restricted) か推奨 (Best Practice) かどうかの補足情報が与えられている。制限は、メタデータ要素が列挙されたボキャブラリに制限されていることを示す。推奨は、ベストプラクティスのために推奨としてリストされたボキャブラリを使用することを SCORM が推薦していることを示している。

14841.12.1-2002: 情報モデル				
項番	名称	説明	繰り返し	データ型
1.	一般 (General)	このカテゴリは学習オブジェクトを総括的に記述する一般的な情報を集めている。	0 or 1	-
1.1.	識別子 (Identifier)	この学習オブジェクトを識別する世界的にユニークなラベル。	0 以上 (最低限保証すべき最大値: 10)	-

1.1.1.	カタログ (Catalog)	このエントリの識別空間またはカタログスキーマの名前もしくは指定子．名前空間スキーマ．	0 または 1	任意文字列 (最低限保証すべき最大値：1000 文字)
1.1.2.	エントリ (Entry)	この学習オブジェクトを指名もしくは識別する識別空間もしくはカタログ化スキーマ内の識別子の値．名前空間固有の文字列．	0 または 1	任意文字列 (最低限保証すべき最大値：1000 文字)
1.2.	タイトル (Title)	学習資源に与えられた名前．	0 または 1	言語文字列 (最低限保証すべき最大値：1000 文字)
1.3.	言語 (Language)	想定されるユーザが学習オブジェクトを利用する際に使用する第 1 言語．	0 以上 (最低限保証すべき最大値：10)	任意文字列 (最低限保証すべき最大値：100 文字)
1.4.	内容説明 (Description)	学習資源の内容についての文章記述．	0 以上 (最低限保証すべき最大値：10)	言語文字列 (最低限保証すべき最大値：2000 文字)
1.5.	キーワード (Keyword)	この学習オブジェクトの属性や特徴を記述するキーワード，またはフレーズ． このデータ要素は，他の要素で記述できる特徴に使用してはならない．	0 以上 (最低限保証すべき最大値：10)	言語文字列 (最低限保証すべき最大値：1000 文字)
1.6.	対象範囲 (Coverage)	この学習オブジェクトが適用される時間・文化・地理・分野． 学習資源オブジェクトの内容の広さや範囲．対象範囲は，典型的に空間の位置（場所名か地理的な座標），一時的な期間（期間ラベル，日付，あるいは日付範囲），区域（指定された管理上の実体のような）などを含む．推奨は，管理された語彙から値を選ぶこと（例えば，[TGN]のような地名シソーラス），特定の座標セットもしくは日付範囲のように，数値の識別が優先して使用されるような場所や期間．	0 以上 (最低限保証すべき最大値：10)	言語文字列 (最低限保証すべき最大値：1000 文字)

1.7.	基本構造 (Structure)	この学習オブジェクトの基礎になっている組織構造。 IEEE LOM ボキャブラリ: <ul style="list-style-type: none"> • アトム型(atomic) - (このコンテキストの中で) 分割不可能なオブジェクト。 • 集積型(collection) - オブジェクトのセットで、それらの中で特殊な関連を持たないもの。 • ネットワーク型(networked) - オブジェクトのセットで、特殊でない関連を持つもの。 • 階層型(hierarchical) - オブジェクトのセットで、木構造で表現できる関連を持つもの。 • 直線型(linear) - オブジェクトのセットで、すべて並んでいるもの。 例: 「前へ」や「次へ」の関連で接続しているオブジェクトのセット。 	0または1	ボキャブラリ (状態) (ADL 注釈: 制限)
1.8.	アグリゲーションレベル (Aggregation Level)	この学習オブジェクトの機能的粒度 IEEE LOM ボキャブラリ: <ul style="list-style-type: none"> • 1: 最小レベル。例えば、生のメディアデータや断片。 • 2: レベル1の学習オブジェクトの集合。例えば、1レッスン。 • 3: レベル2の学習オブジェクトの集合。例えば、1コース。 • 4: 粒度の最大レベル。例えば、証明書に結びつくコースのセット。 	0または1	ボキャブラリ (数値) (ADL 注釈: 制限)
2.	ライフサイクル (Life Cycle)	このカテゴリは、この学習オブジェクトの経歴や現状、及び発展の過程で影響を与えた実体を記述する。	0または1	-
2.1.	バージョン (Version)	この学習オブジェクトのバージョン。	0または1	言語文字列 (最低限保証すべき最大値: 50文字)
2.2.	ステータス (Status)	この学習オブジェクトの完了状態または条件 IEEE LOM ボキャブラリ: <ul style="list-style-type: none"> • ドラフト(draft) • ファイナル(final) • 改訂版(revised) • 利用不可(unavailable) 	0または1	ボキャブラリ (状態) (ADL 注釈: 制限)

2.3.	コントリビュート (Contribute)	この学習オブジェクトのライフサイクル中（作成・編集・出版など）に貢献したもの（人物や組織）。	0 以上 （最低限保証すべき最大値：30）	-
2.3.1.	役割 (Role)	貢献（コントリビュート）の種類 IEEE LOM ポキャブラリ: <ul style="list-style-type: none"> • 製作者(author) • 発行者(publisher) • 不明(unknown) • 発案者(initiator) • 完成者(terminator) • 検証者(validator) • 編集者(editor) • グラフィックデザイナー(graphical designer) • 技術的実装者(technical implementer) • コンテンツ提供者(content provider) • 技術的検証者(technical validator) • 教育的検証者(educational validator) • 執筆者(script writer) • インストラクショナルデザイナー(instructional designer) • 領域専門家(subject matter expert) 	0 または 1	ポキャブラリ（状態） （ADL 注釈: 推奨）
2.3.2.	情報 (Entity)	この学習オブジェクトに貢献（コントリビュート）したものの（人物や組織）についての情報や識別子。最も関係の深いものから順に記述する。	0 以上 （最低限保証すべき最大値：40）	任意文字列 （最低限保証すべき最大値：1000 文字）
2.3.3.	日付 (Date)	貢献した日付。	0 または 1	日付形式
3.	メタメタデータ (Meta-Metadata)	このカテゴリは、このメタデータレコード自身を記述する。（学習オブジェクトに関してではない） このカテゴリは、どのようにメタデータのインスタンスを認識でき、誰がいつどうやって何を参照してこのメタデータを生成したかということを記述する。	0 または 1	-
3.1.	識別子 (Identifier)	このメタデータレコードを識別する世界的にユニークなラベル。	0 以上 （最低限保証すべき最大値：10）	-

3.1.1.	カタログ (Catalog)	このエントリの識別空間またはカタログスキーマの名前もしくは指定子．名前空間スキーマ．	0 または 1	任意文字列 (最低限保証すべき最大値：1000 文字)
3.1.2.	エントリ (Entry)	このメタデータレコードを指名もしくは識別する識別空間もしくはカタログ化スキーマ内の識別子の値．名前空間固有の文字列．	0 または 1	任意文字列 (最低限保証すべき最大値：1000 文字)
3.2.	コントリビュート (Contribute)	このメタデータインスタンスのライフサイクル中（作成・検証など）に貢献したものの（人物や組織）．	0 以上 (最低限保証すべき最大値：10)	-
3.2.1.	役割 (Role)	貢献（コントリビュート）の種類． 「作成者」のインスタンスは唯一であることを推奨する． IEEE LOM ボキャブラリ: <ul style="list-style-type: none"> • 作成者(creator) • 検証者(validator) 	0 または 1	ボキャブラリ（状態） (ADL 注釈: 推奨)
3.2.2.	情報 (Entity)	このメタデータインスタンスに貢献（コントリビュート）したものの（人物や組織）についての情報や識別子．最も関係の深いものから順に記述する．	0 以上 (最低限保証すべき最大値：10)	任意文字列 (最低限保証すべき最大値：1000 文字)
3.2.3.	日付 (Date)	貢献した日付．	0 または 1	日付形式
3.3.	メタデータスキーマ (Metadata Scheme)	このメタデータインスタンスの生成に使用した仕様の名前やバージョン．	0 以上 (最低限保証すべき最大値：10)	任意文字列 (最低限保証すべき最大値：30 文字)
3.4.	言語 (Language)	メタデータインスタンスの言語．これは、このメタデータインスタンスにおけるすべての言語文字列の値に対する既定の言語である．このデータ要素に対する値がメタデータインスタンスの中にある場合、言語文字列値の既定言語は設定されない．	0 または 1	任意文字列 (最低限保証すべき最大値：100 文字)
4.	技術的事項 (Technical)	このカテゴリは、この学習オブジェクトの技術的な要求事項や特性を記述する．	0 または 1	-

4.1.	フォーマット (Format)	この学習オブジェクト（のすべての構成要素）の技術的なデータタイプ（複数）.. この要素は、学習オブジェクトへのアクセスに必要なソフトウェアの識別に使用される。	0 以上 (最低限保証すべき最大値：40)	任意文字列 (最低限保証すべき最大値：500 文字)
4.2.	サイズ (Size)	デジタル学習オブジェクトのバイト数（8bit）。サイズは 10 進数（根 10）として表現する。したがって、'0'..'9'の数字で表現され、単位は bytes である。 (Mbytes,GB 等ではない) このデータ要素は、学習オブジェクトの実際のサイズを参照する。もし学習オブジェクトが圧縮されていたら、このデータ要素は圧縮前のサイズを参照する。	0 または 1	任意文字列 (最低限保証すべき最大値：30 文字)
4.3.	格納場所 (Location)	この学習オブジェクトにアクセスするために使用する文字列。Universal Resource Locator (URL) 等の格納場所や、Universal Resource Identifier (URI) のような格納場所を探す方法。 このリストの最初の要素は、望ましい格納場所とする。	0 以上 (最低限保証すべき最大値：10)	任意文字列 (最低限保証すべき最大値：1000 文字)
4.4.	必要条件 (Requirement)	この学習オブジェクトを使用する場合の技術的な要件。 複数の必要条件がある場合は、すべてが要求される。つまり、論理接続詞は AND である。	0 以上 (最低限保証すべき最大値：40)	-
4.4.1.	複数条件 (OrComposite)	多数の必要条件を複合したもの。複数の必要条件は、構成要素の必要条件のうち 1 つが満たされるということである。つまり、論理接続詞は OR である。	0 以上 (最低限保証すべき最大値：40)	-
4.4.2.	タイプ (Type)	この学習オブジェクトを使用する際に要求される技術。つまり、ハードウェア、ソフトウェア、ネットワーク等である。 IEEE LOM ポキャブラリ: <ul style="list-style-type: none"> • OS(operating system) • ブラウザ(browser) 	0 または 1	ポキャブラリ (状態) (ADL 注釈: 推奨)

4.4.3.	名前 (Name)	この学習オブジェクトを使用する場合に要求される技術の名前。 IEEE LOM ボキャブラリ: <ul style="list-style-type: none">• タイプが“operating system”の場合<ul style="list-style-type: none">○ pc-dos○ ms-windows○ macos○ unix○ multi-os○ none• タイプが“browser”の場合<ul style="list-style-type: none">○ any○ netscape communicator○ ms-internet explorer○ opera○ amaya	0 または 1	ボキャブラリ (状態) (ADL 注釈: 推奨)
4.4.4.	最低バージョン (Minimum Version)	この学習オブジェクトを使用する際に要求される技術の最低限可能なバージョン。	0 または 1	任意文字列 (最低限保証すべき最大値: 30 文字)
4.4.5.	最高バージョン (Maximum Version)	この学習オブジェクトを使用する際に要求される技術の最上限可能なバージョン。	0 または 1	任意文字列 (最低限保証すべき最大値: 30 文字)
4.5.	導入方法 (Installation Remarks)	この学習オブジェクトのインストール方法に関する記述。	0 または 1	言語文字列 (最低限保証すべき最大値: 1000 文字)
4.6.	その他のシステム要件 (Other Platform Requirements)	その他のソフトウェア, ハードウェアの要求に関する情報。	0 または 1	言語文字列 (最低限保証すべき最大値: 1000 文字)
4.7.	継続時間 (Duration)	学習オブジェクトを想定された速度で再生した場合に要する時間。	0 または 1	期間形式
5.	教育的事項 (Educational)	このカテゴリは, 学習オブジェクトのキーとなる教育的または教育学的特性を記述する。	0 または 1 0 以上 (最低限保証すべき最大値: 100)	-

5.1.	情報交換タイプ (Interactivity Type)	<p>この学習オブジェクトでサポートされた，学習の顕著な形態．</p> <p>IEEE LOM ボキャブラリ:</p> <ul style="list-style-type: none"> • 対話型 (active) • 提示型 (expositive) • 対話 / 提示混合型 (mixed) <p>“対話型”の学習（行動による学習）は，学習者に生産的な行為を直接引き起こすコンテンツにサポートされる．対話型の学習オブジェクトは，意味論的に有意義な入力のため，あるいは他のある種の生産的な行為または決定（必ずしも学習オブジェクトのフレームワーク内で実行されなくても）のために学習者を促す．対話型ドキュメントはシミュレーション，問題集，およびドリルなどである．</p> <p>“提示型”の学習（受動的な学習）は，学習者の仕事が，主に提示されたコンテンツを吸収することから成る場合に発生する（一般にテキスト，画像，音声による）．提示型の学習オブジェクトは情報は表示するが，意味論的に有意義な入力を学習者に促さない．提示型ドキュメントはエッセイ，ビデオクリップ，すべての種類の画像教材，ハイパーテキストドキュメントなどである．</p> <p>学習オブジェクトが，対話型と提示型が相互作用するようブレンディングしたタイプの場合，その相互作用のタイプは“対話 / 提示混合型”である．</p>	0 または 1	ボキャブラリ（状態） （ADL 注釈: 制限）
------	---------------------------------	--	---------	----------------------------

5.2.	学習資源タイプ (Learning Resource Type)	<p>学習オブジェクトの主要な種類．最も関係の深いものを最初に記述すべきである．</p> <p>ボキャブラリは，<i>学習資源</i>の特定の目的に適応されている．</p> <p>IEEE LOM ボキャブラリ:</p> <ul style="list-style-type: none"> • ドリル (exercise) • シミュレーション (simulation) • 問題集 (questionnaire) • 図形 (diagram) • 画像 (figure) • グラフ (graph) • インデックス (index) • スライド (slide) • 表 (table) • 説明文 (narrative text) • 試験 (exam) • 実験 (experiment) • 問題記述 (problem statement) • 自己評価 (self assessment) • 講義 (lecture) 	0 以上 (最低限保証すべき最大値：10)	ボキャブラリ (状態) (ADL 注釈: 推奨)
5.3.	対話性 (Interactivity Level)	<p>この学習オブジェクトを特徴付ける相互作用の程度．このコンテキストの相互作用は，学習オブジェクトの様相や動作に，学習者が影響を及ぼすことができる程度を参照する．</p> <p>IEEE LOM ボキャブラリ:</p> <ul style="list-style-type: none"> • 大変低い (very low) • 低い (low) • 普通 (medium) • 高い (high) • 大変高い (very high) 	0 または 1	ボキャブラリ (数値) (ADL 注釈: 制限)

5.4.	学習効率 (Semantic Density)	<p>学習オブジェクトの有効性の程度．学習オブジェクトの意味的な密度は，そのサイズや範囲，（オーディオやビデオのように限られた時間のものは）継続時間などで判断される．</p> <p>学習オブジェクトの意味的な密度は，その難易度には依存しない．提示型の素材はよい例であるが，対話型の資源でも同様に使用される．</p> <p>IEEE LOM ポキャブラリ:</p> <ul style="list-style-type: none"> • 大変低い (very low) • 低い (low) • 普通 (medium) • 高い (high) • 大変高い (very high) 	0 または 1	<p>ポキャブラリ (数値)</p> <p>(ADL 注釈: 制限)</p>
5.5.	利用者の種別 (Intended End User Role)	<p>この学習オブジェクトの設計時に想定されている主要な利用者 (複数)．もっとも関係の深いものを最初に記述する．</p> <p>IEEE LOM ポキャブラリ:</p> <ul style="list-style-type: none"> • 教師 (teacher) • 製作者 (author) • 学習者 (learner) • 管理者 (manager) 	0 以上 (最低限保証すべき最大値: 10)	<p>ポキャブラリ (状態)</p> <p>(ADL 注釈: 制限)</p>
5.6.	利用環境 (Context)	<p>学習や，この学習オブジェクトの利用が想定される典型的な環境．</p> <p>IEEE LOM ポキャブラリ:</p> <ul style="list-style-type: none"> • 学校 (school) • 高等教育 (higher education) • 訓練 (training) • その他 (other) 	0 以上 (最低限保証すべき最大値: 10)	<p>ポキャブラリ (状態)</p> <p>(ADL 注釈: 推奨)</p>
5.7.	主な年齢層 (Typical Age Range)	<p>想定されるユーザの年齢層．</p> <p>このデータ要素は，発達年齢 (developmental age) であり，年齢 (chronological age) ではない．</p> <p>できれば，(x 才 ~ y 才) (z 才 ~) の形式にすべきである．「注釈: これは，3つの構成要素 (最低年齢・最高年齢・説明記述) を置くか，フリーのテキストフィールドを置くだけにするかの折衷案となる．」</p>	0 以上 (最低限保証すべき最大値: 5)	<p>言語文字列</p> <p>(最低限保証すべき最大値: 1000 文字)</p>

5.8.	難易度 (Difficulty)	この学習オブジェクトを、標準的な対象ユーザが実行し通過するのがどのくらい難しいか。 IEEE LOM ボキャブラリ: <ul style="list-style-type: none"> • 大変易しい (very easy) • 易しい (easy) • 標準レベル (medium) • 難しい (difficult) • 大変難しい (very difficult) 	0 または 1	ボキャブラリ (数値) (ADL 注釈: 制限)
5.9.	学習時間 (Typical Learning Time)	この学習オブジェクトを、標準的な対象ユーザが実行し通過するのにかかる大体の、または標準的な時間。	0 または 1	期間形式
5.10.	記述 (Description)	この学習オブジェクトをどのように使用するかについてのコメント。	0 以上 (最低限保証すべき最大値: 10)	言語文字列 (最低限保証すべき最大値: 1000 文字)
5.11.	言語 (Language)	標準的な想定ユーザが使用する言語。	0 以上 (最低限保証すべき最大値: 10)	任意文字列 (最低限保証すべき最大値: 100 文字)
6.	権利 (Rights)	このカテゴリは、学習オブジェクトの著作権や使用条件について記述する。	0 または 1	-
6.1.	費用 (Cost)	学習オブジェクトを使用するために使用料が求められるかどうか。 IEEE LOM ボキャブラリ: <ul style="list-style-type: none"> • yes • no 	0 または 1	ボキャブラリ (状態) (ADL 注釈: 制限)
6.2.	著作権および制約 (Copyright and Other Restrictions)	著作権や他の制約がこの学習オブジェクトの使用に適用されるかどうか。 IEEE LOM ボキャブラリ: <ul style="list-style-type: none"> • yes • no 	0 または 1	ボキャブラリ (状態) (ADL 注釈: 制限)
6.3.	記述 (Description)	この学習オブジェクトの使用条件についてのコメント。	0 または 1	言語文字列 (最低限保証すべき最大値: 1000 文字)

7.	他のオブジェクトとの関連 (Relation)	このカテゴリは、この学習オブジェクトと他の学習オブジェクトの関係を定義している。 複数の関係を定義する場合は、複数のインスタンスを持たせることができる。もし、対象の学習オブジェクトが複数の場合は、各対象について新しく関係付けのインスタンスを定義する。	0 以上 (最低限保証すべき最大値：100)	-
7.1.	種別 (Kind)	この学習オブジェクトと 7.2:Relation.Resource によって識別される対象の学習オブジェクトとの関係。 IEEE LOM ポキャブラリ: <ul style="list-style-type: none"> • 包含元 (ispartof) • 包含 (haspart) • バージョン元 (isversionof) • 一つのバージョン (hasversion) • 形式元 (isformatof) • 一つの形式 (hasformat) • 参照先 (references) • 参照元 (isreferencedby) • 基礎 (isbasedon) • 応用 (isbasisfor) • 必須 (requires) • 要求元 (isrequiredby) 	0 または 1	ポキャブラリ (状態) (ADL 注釈: 推奨)
7.2.	資源 (Resource)	この関係の対象となる学習オブジェクト。	0 または 1	-
7.2.1.	識別子 (Identifier)	対象となる学習オブジェクトを識別する、世界的にユニークな識別子。	0 以上 (最低限保証すべき最大値：10)	-
7.2.1.1.	カタログ (Catalog)	このエントリの識別空間またはカタログスキーマの名前もしくは指定子。名前空間スキーマ。	0 または 1	任意文字列 (最低限保証すべき最大値：1000 文字)
7.2.1.2.	エントリ (Entry)	この学習オブジェクトを指名もしくは識別する識別空間もしくはカタログ化スキーマ内の識別子の値。名前空間固有の文字列。	0 または 1	任意文字列 (最低限保証すべき最大値：1000 文字)

7.2.2.	記述 (Description)	対象となる学習オブジェクトの記述 .	0 以上 (最低限保証すべき最大値 : 10)	言語文字列 (最低限保証すべき最大値 : 1000 文字)
8.	注釈 (Annotation)	このカテゴリは、この学習オブジェクトの教育的使用におけるコメントや、コメントがいつ誰によって作成されたかという情報を供給する . このカテゴリは、教育者に学習オブジェクトの評価や使用のための提案を共有させる .	0 以上 (最低限保証すべき最大値 : 30)	-
8.1.	情報 (Entity)	この注釈を作成したもの (人物や組織) .	0 または 1	任意文字列 (最低限保証すべき最大値 : 1000 文字)
8.2.	日付 (Date)	この注釈を作成した日付 .	0 または 1	日付形式
8.3.	記述 (Description)	この注釈の内容 .	0 または 1	言語文字列 (最低限保証すべき最大値 : 1000 文字)
9.	分類体系 (Classification)	このカテゴリは、特定の分類規則 (システム) でどこにこの学習オブジェクトが属するかを記述する . 複数の分類を定義する場合は、このカテゴリに複数のインスタンスを置くことができる .	0 以上 (最低限保証すべき最大値 : 40)	-
9.1.	目的 (Purpose)	この学習オブジェクトの分類目的 . IEEE LOM ポキャプラリ: <ul style="list-style-type: none">• 教科 (discipline)• アイデア (idea)• 必要条件 (prerequisite)• 学習目的 (educational objective)• アクセス可能性 (accessibility)• 制限 (restrictions)• 教育的レベル (educational level)• スキルレベル (skill level)• セキュリティレベル (security level)• コンピテンシ (competency)	0 または 1	ポキャプラリ (状態) (ADL 注釈: 推奨)

9.2.	分類体系へのパス (Taxon Path)	<p>特定の分類規則（システム）における分類体系パス（＝タクソノミ）。後ろに続くレベルはそれぞれ、より高いレベルにおける細かい定義である。</p> <p>同じまたは異なる分類における異なるパスが同じ特性を記述する場合もありうる。</p> <p>タクソノミは、言葉やフレーズ（taxon）の階層である。</p>	0以上 (最低限保証すべき最大値：15)	-
9.2.1.	分類体系の名前 (Source)	<p>分類体系（システム）の名前。</p> <p>この要素は、"公式"タクソノミまたはユーザが定義したタクソノミの識別に使用される。</p>	0または1	言語文字列 (最低限保証すべき最大値：1000文字)
9.2.2.	各分類項目 (Taxon)	<p>タクソノミで用いられる特定の用語。各分類項目（taxon）はラベルや用語の定義されたノードである。各分類項目（taxon）はまた、標準化された参照のための英数字の名称、あるいは識別子を持つ。ラベルやエントリは、特定の各分類項目（taxon）を示すために使用される。</p> <p>各分類項目（taxon）の順番付きリスト（ordered list）は、分類体系へのパス（taxonomic stairway）を生成する。これは、一般的なエントリから特徴的なエントリへのパスである。</p>	0以上 (最低限保証すべき最大値：15)	-
9.2.2.1.	識別子 (Id)	<p>数字や文字の組み合わせである各分類項目の識別子は、タクソノミのソースによって与えられる。</p>	0または1	言語文字列 (最低限保証すべき最大値：100文字)
9.2.2.2.	エントリ (Entry)	<p>各分類項目のテキストラベル</p>	0または1	言語文字列 (最低限保証すべき最大値：500文字)
9.3.	記述 (Description)	<p>この特定の分類体系における 9.1:Purpose の分類項目の指定（教科、アイデア、スキルレベル、学習目的など）と学習オブジェクトの関係についての記述。</p>	0または1	言語文字列 (最低限保証すべき最大値：2000文字)

9.4.	キーワード (Keyword)	この特定の分類体系における 9.1:Purpose の分類項目の指定（アクセス可能性，セ キュリティレベルなど）と学習オブジェ クトの関係を表すキーワードやフレー ズ．もっとも関係の深いものを最初に記 述する．	0 以上 (最低限保 証すべき最 大値：40)	言語文字列 (最低限保証すべ き最大値：1000 文 字)
------	--------------------	--	--------------------------------------	--

DRAFT

LangString 型の情報モデル				
項番	名称	説明	繰り返し	データ型
1	言語文字列 (Langstring)	一つ以上の文字列をあらわすデータ型． 言語文字列の値は，翻訳あるいは代替記述のような多数の意味論的に等価な文字列を含んでいる．	0 以上 (最低限保証すべき最大値：10)	-
1.1	言語 (Language)	文字列の言葉の言語	0 または 1	任意文字列 (最低限保証すべき最大値：100 文字)
1.2	文字列 (String)	実際の文字列の値．	0 または 1	任意文字列 (CharacterString)

日付形式の情報モデル				
項番	名称	説明	繰り返し	データ型
1	日付形式 (DateTime)	少なくとも 1 秒と同じくらい小さい精度を備えた時間． YYYY[-MM][-DD[Thh[:mm[:ss[:s[TZD]]]]]] 位置: YYYY = 年 4 桁 (>=0001) MM = 月 2 桁 (01 ~ 12) DD = 日 2 桁 (01 ~ 31) hh = 時 2 桁 (00 ~ 23) mm = 分 2 桁 (00 ~ 59) ss = 秒 2 桁 (00 ~ 59) s = 秒の小数を 1 桁かそれ以上で表現する 10 進数 TZD = 時間帯指定 (time zone designator) 少なくとも，年は 4 桁存在しなければならない．日付形式の追加の部分が含まれる場合，文字列リテラル “-”，“T”，“.”，“.”，は日付形式 (datetime) の一部である．	0 または 1	任意文字列 (最低限保証すべき最大値：200 文字)

2	記述 (Description)	日付の説明 .	0 または 1	言語文字列 (最低限保証すべき 最大値 : 1000 文字)
---	---------------------	---------	---------	--

期間 (Duration) 型の情報モデル				
項番	名称	説明	繰り返し	データ型
1	期間 (Duration)	少なくとも 1 秒と同じくらい小さい精度を備えた時間の間隔 . P[yY][mM][dD][T[hH][nM][s[.s]S]] 位置: y = 年 m = 月 d = 日 h = 時 n = 分 s = 秒または秒の小数 文字列リテラル “P”, “Y”, “M”, “D”, “T”, “H”, “M”, “S” は, それぞれに対応する 0 以外の値が存在する場合, 表記しなければならない .	0 または 1	任意文字列 (最低限保証すべき 最大値 : 200 文字)
2	記述 (Description)	期間の説明 .	0 または 1	言語文字列 (最低限保証すべき 最大値 : 1000 文字)

ボキャブラリの情報モデル				
項番	名称	説明	繰り返し	データ型
1	ソース (Source)	“LOMv1.0”, もしくは URI のような値の出典の表示 .	0 または 1	任意文字列 (最低限保証すべき 最大値 : 1000 文字)
2	値 (Value)	実際の値 出典が “LOMv1.0” である場合, 値はデータ要素用の LOMv1.0 基本スキーマに定義されたリストによるものとする .	0 または 1	任意文字列 (最低限保証すべき 最大値 : 1000 文字)

3.1.3. SCORM メタデータ XML バインディング

SCORM メタデータ XML バインディングは、学習オブジェクトメタデータ (LOM) XML バインディングのための IEEE P1484.12.3/D2, 2003-01-17 ドラフト標準に基づいている。XML バインディングは、SCORM メタデータ情報モデルがどのように解釈されて XML に割り当てられるかを定義している。バインディングは、情報モデルの項目のどの項目がどの XML 要素や XML 属性等に対応付けられるかを示すことによって、情報モデルの項目がどのように XML で表現されるかを説明する。

いくつかの要素には、要素リスト中のデータ要素や任意文字列データ型、言語文字列データ型のデータ要素に対して「最低限保証されるべき数」が書かれている。これは、すべてのアプリケーションは少なくともその数のリストのエントリ、またはその長さの文字列（任意文字列や言語文字列）をサポートすることを示す。

3.1.3.1 <lom> 要素

説明: 最も外側のルート of 要素。この要素は、SCORM メタデータ XML レコードの始まりを示す。

型: 親要素

繰り返し: <lom>要素は XML メタデータインスタンスのルート要素である。この要素は、SCORM XML メタデータ文書中にちょうど 1 回出現しなくてはならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

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

3.1.3.1.1 <general>要素

説明: <general>要素グループは, その学習オブジェクトの属性や特徴を総括する一般的情報を記述する.

型: 親要素

繰り返し: <general>要素は, トップレベルの<lom>要素の中に1回のみ出現しなくてはならない.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1回のみ
アクティビティ	1回のみ
SCO	1回のみ
SCA	1回のみ
アセット	1回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

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

3.1.3.1.1.1 <identifier>要素

説明: <identifier>要素は, この学習オブジェクトを識別する世界的にユニークなラベルを表す.

型: 親要素

繰り返し: <identifier>要素は<general>要素中に1回以上出現する. 最低限保証すべき最大値は10項目である.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1回以上
アクティビティ	1回以上
SCO	1回以上
SCA	1回以上
アセット	1回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- <catalog>
- <entry>

例:

```
<identifier>
  <catalog>Sample Catalog 1</catalog>
  <entry>Sample Entry 1</entry>
</identifier>
```

3.1.3.1.1.1.1. <catalog> 要素

Description: <catalog>要素は、このエントリの識別空間またはカタログスキーマの名前もしくは指定子を記述する。

型: 任意文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <catalog>要素は、<identifier>要素中に 1 回のみ出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<identifier>
  <catalog>Sample Catalog 1</catalog>
  <entry>Sample Entry 1</entry>
</identifier>
```

3.1.3.1.1.1.2. <entry>要素

説明: <entry>要素は、識別空間もしくはカタログ化スキーマ内の識別子の値を定義する。

型: 任意文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <entry>要素は、<identifier>要素中に 1 回のみ出現しなくてはならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ

アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<identifier>
  <catalog>Sample Catalog 1</catalog>
  <entry>Sample Entry 1</entry>
</identifier>
```

3.1.3.1.1.2 <title>要素

説明: <title>要素は、学習オブジェクトに与えられた名前を記述する。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <title>要素は、<general>要素中に 1 回のみ出現しなくてはならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<general>
  <title>
    <string language="en">Title 1 in English</string>
    <string language="fr">Title 1 in French</string>
  </title>
</general>
```

3.1.3.1.1.3 <language>要素

説明: <language>要素は、第 1 言語、あるいは想定ユーザがこの学習オブジェクトでやりとりするのに使用する言語を記述する。

型: 任意文字列 (最低限保証すべき最大値: 100 文字)

繰り返し: <language>要素は<general>要素中に 1 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

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

3.1.3.1.1.4 <description>要素

説明: <description>要素には、この学習オブジェクトの内容についての文章記述を書く。

型: 言語文字列 (最低限保証すべき最大値: 2000 文字)

繰り返し: <description>要素は、<general>要素中に 1 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	1 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<general>
```

```

<description>
  <string language="en">Description in English</string>
  <string language="fr">Description in French</string>
</description>
</general>

```

3.1.3.1.1.5 <keyword>要素

説明: <keyword>要素は、この学習オブジェクトの属性や特徴を表すキーワードやフレーズを記述する。このデータ要素は、他のデータ要素で記述できない特性に使用されるべきである。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <keyword>要素は<general>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	1 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```

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

```

3.1.3.1.1.6 <coverage>要素

説明: <coverage>要素は、この学習オブジェクトが適用される時間・文化・地理・分野等を記述する。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <coverage>要素は<general>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

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

3.1.3.1.1.7 <structure>要素

説明: <structure>要素は、この学習オブジェクトの基礎になっている組織構造を記述する。

型: ボキャブラリ

繰り返し: <structure>要素は、<general>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- アトム型 (atomic) - (このコンテキストの中で) 分割不可能なオブジェクト
- 集積型 (collection) - オブジェクトのセットで , それらの間で特殊な関係を持たないもの
- ネットワーク型 (networked) - オブジェクトのセットで , 特殊でない関連をもつもの
- 階層型 (hierarchical) - オブジェクトのセットで , 木構造で表現できる関連をもつもの
- 直線型 (linear) - オブジェクトのセットで , すべて並んでいるもの .
例 : 「前へ」や「次へ」の関連で接続しているオブジェクトのセット

例:

```
<general>
  <structure>
    <source>LOMv1.0</source>
    <value>collection</value>
  </structure>
</general>
```

3.1.3.1.1.8 <aggregationLevel> 要素

説明: <aggregationLevel>要素は , この学習オブジェクトの機能的な粒度を記述する .

型: ボキャブラリ

繰り返し: <aggregationlevel>要素は , <general>要素中に 0 回または 1 回出現する .

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 - 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 1: アグリゲーションの最小レベル . 例えば , 生のメディアデータや断片
- 2: レベル 1 の学習オブジェクトの集合 . 例えば , 1 レッスン
- 3: レベル 2 の学習オブジェクトの集合 . 例えば , 1 コース
- 4: 粒度の最大レベル . 例えば , 証明書に結びつくコースのセット

例:

```
<general>
  <aggregationLevel>
    <source>LOMv1.0</source>
    <value>1</value>
  </aggregationLevel>
</general>
```

3.1.3.1.2 <lifeCycle>要素

説明: <lifeCycle>要素は、この学習オブジェクトの経歴や現状、製作に貢献した人物等を記述する。

型: 親要素

繰り返し: <lifeCycle>要素は、トップレベルの<lom>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

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

3.1.3.1.2.1 <version>要素

記述: <version>要素は、この学習オブジェクトのバージョンを記述する。

型: 言語文字列 (最低限保証すべき最大値: 50 文字)

繰り返し: <version>要素は、<lifeCycle>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

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

3.1.3.1.2.2 <status>要素

説明: <status>要素は、この学習オブジェクトの完全な状態・状況を記述する。

型: ボキャブラリ

繰り返し: <status>要素は、<lifeCycle>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- ドラフト (draft)
- ファイナル (final)
- 改訂版 (revised)
- 入手不可 (unavailable)

例:

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

3.1.3.1.2.3 <contribute>要素

説明: <contribute>要素は、この学習オブジェクトのライフサイクル中（作成・編集・出版など）に貢献したもの（人物や組織）を記述する。

型: 親要素

繰り返し: <contribute>要素は、<lifecycle>要素中に 0 回以上出現する。最低限保証すべき最大値は 30 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <role>
- <entity>
- <date>

例:

```
<lifecycle>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>author</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Administrator\nAssistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</lifecycle>
```

3.1.3.1.2.3.1. <role>要素

説明: <role>要素は、貢献（コントリビュート）の種類を記述する。

型: ボキャブラリ

繰り返し: <contribute>要素が使われている場合, <role>要素は, <contribute>要素中に 0 回または 1 回出現する. 複数の貢献者が (異なった役割で) 存在する場合, <contribute>要素は, 繰り返されるべきである.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 製作者 (author)
- 発行者 (publisher)
- 不明 (unknown)
- 発案者 (initiator)
- 完成者 (terminator)
- 検証者 (validator)
- 編集者 (editor)
- グラフィックデザイナー (graphical designer)
- 技術的実装者 (technical implementer)
- コンテンツ提供者 (content provider)
- 技術的検証者 (technical validator)
- 教育的検証者 (educational validator)
- 執筆者 (script writer)
- インストラクショナルデザイナー (instructional designer)
- 領域専門家 (subject matter expert)

例:

```
<lifeCycle>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>author</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Administrator\nAssistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</lifeCycle>
```

3.1.3.1.2.3.2. <entity>要素

説明. <entity>要素は、この学習オブジェクトに貢献（コントリビュート）した
もの（人物や組織）についての情報や識別子を提供する。この構成要素は、最
も関係の深いものから順に記述する。

型: VCard（最低限保証されるべき数: 1000 文字）

繰り返し: <entity>要素は、<contribute>要素中に 0 回以上出現する。最低限保証
すべき最大値は 40 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: VCard (VCard Type) を参照

例:

```
<lifeCycle>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>author</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Administrator\nAssistant\nEMAIL\;/TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</lifeCycle>
```

3.1.3.1.2.3.3. <date> 要素

説明: この<date>要素は、貢献した日付を記述する。

型: 日付形式

繰り返し: <date>要素は、<contribute>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 日付形式 (Date Time Type) を参照

例:

```
<lifeCycle>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>author</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Administrator\nAssistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</lifeCycle>
```

3.1.3.1.3 <metaMetadata>要素

説明: <metaMetadata>要素は、このメタデータレコード自身を記述する。(学習オブジェクトに関してではない)

型: 親要素

繰り返し: <metaMetadata>要素は、トップレベルの<lom>要素中に1回のみ出現しなければならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1回のみ
アクティビティ	1回のみ
SCO	1回のみ
SCA	1回のみ
アセット	1回のみ

SCORM アプリケーションプロファイル要求事項

要素:

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

3.1.3.1.3.1 <identifier>要素

説明: <identifier>要素は、このメタデータレコードを識別する世界的にユニークなラベルを記述する。

型: 親要素

繰り返し: <identifier>要素は, <metaMetadata>要素中に 1 回以上出現する. 最低限保証すべき最大値は 10 項目である.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	1 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <catalog>
- <entry>

例:

```
<identifier>  
  <catalog>Sample Catalog 1</catalog>  
  <entry>Sample Entry 1</entry>  
</identifier>
```

3.1.3.1.3.1.1. <catalog>要素

説明: <catalog>要素は, このエントリの識別空間またはカタログスキーマの名前もしくは指定子を記述する.

型: 任意文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <catalog>要素は, <identifier>要素中に 1 回のみ出現しなければならない.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<identifier>
  <catalog>Sample Catalog 1</catalog>
  <entry>Sample Entry 1</entry>
</identifier>
```

3.1.3.1.3.1.2. <entry>要素

説明: <entry>要素は、識別空間もしくはカタログ化スキーマ内の識別子の値を定義する。

型: 任意文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <entry>要素は、<identifier>要素中に 1 回のみ出現しなければならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<identifier>
  <catalog>Sample Catalog 1</catalog>
  <entry>Sample Entry 1</entry>
</identifier>
```

3.1.3.1.3.2 <contribute>要素

説明: <contribute>要素は、このメタデータインスタンスのライフサイクル中 (作成・検証など) に貢献したもの (人物や組織) を記述する。

型: 親要素

繰り返し: <contribute>要素は、<metaMetadata>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上

SCA	0回以上
アセット	0回以上

SCORM アプリケーションプロファイル要求事項

要素:

- <role>
- <entity>
- <date>

Example:

```

<metaMetadata>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>creator</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Adminstrator\,Assistant\nEMAIL\;/TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</metaMetadata>

```

3.1.3.1.3.2.1. <role> 要素

説明: <role>要素は、貢献の種類を記述する。

型: ボキャブラリ

繰り返し: もし<contribute>要素が使用されるならば、<role>要素は、<contribute>要素中に0回または1回出現する。もし、複数の貢献者が(異なった役割で)存在する場合、<contribute>要素は繰り返される。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0または1
アクティビティ	0または1
SCO	0または1
SCA	0または1
アセット	0または1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 作成者 (creator)
- 検証者 (validator)

例:

```
<metaMetadata>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>creator</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Administrator\nAssistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</metaMetadata>
```

3.1.3.1.3.2.2. <entity>要素

説明: この<entity>要素は、この学習オブジェクトに貢献 (コントリビュート) したものの (人物や組織) についての情報や識別子を記述する。この構成要素は、最も関係が深いものから順に記述する。

型: VCard (最低限保証すべき最大値: 1000 文字)

繰り返し: <entity>要素は、<contribute>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: VCard (VCard Type) を参照

例:

```
<metaMetadata>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>creator</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Administrator\nAssistant\nEMAIL\;/TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</metaMetadata>
```

3.1.3.1.3.2.3. <date>要素

説明: この<date>要素は、貢献した日付を記述する。

型: 日付形式

繰り返し: <date>要素は、<contribute>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 日付形式 (Date Time Type) を参照

例:

```
<metaMetadata>
  <contribute>
    <role>
      <source>LOMv1.0</source>
      <value>creator</value>
    </role>
    <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Administrator\nAssistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
    <date>
      <dateTime>2000-12-12</dateTime>
      <description>
        <string language="en">Date Description</string>
      </description>
    </date>
  </contribute>
</metaMetadata>
```

3.1.3.1.3.3 <metadataSchema>要素

説明: <metadataSchema>は、このメタデータインスタンスを作成するのに用いられた仕様の名称やバージョンを記述する。

型: 任意文字列 (最低限保証すべき最大値: 30 文字)

繰り返し: <metadataSchema>要素は<metaMetadata>要素中に 1 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	1 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<metaMetadata>
  <metadataSchema>ADLv1.3</metadataSchema>
</metaMetadata>
```

3.1.3.1.3.4 <language>要素

説明: <language>要素は、このメタデータインスタンスの言語を記述する。これは、このメタデータインスタンスにおける言語文字列型の要素の既定言語である。このデータ要素に対する値がメタデータインスタンスの中にない場合、言語文字列型の要素に対する既定の言語は設定されない。

型: 任意文字列（最低限保証すべき最大値: 100 文字）

繰り返し: <language>要素は、<metaMetadata>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

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

3.1.3.1.4 <technical>要素

説明: <technical>要素は、この学習オブジェクトの技術的な要求事項や特性を記述する。

型: 親要素

繰り返し: <technical>要素は、トップレベルの<lom>要素中に 1 回だけ出現しなければならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

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

3.1.3.1.4.1 <format>要素

説明: <format>要素は、この学習オブジェクト（を構成する要素すべて）の技術的なデータ型を記述する。このデータ要素は、学習オブジェクトにアクセスするために必要なソフトウェアの特定に用いられる。

型: 任意文字列（最低限保証すべき最大値: 500 文字）

繰り返し: <format>要素は、<technical>要素中に 1 回以上出現しなければならない。最低限保証すべき最大値は 40 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	1 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

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

3.1.3.1.4.2 <size> 要素

説明: <size>要素は、デジタル学習オブジェクトのバイト数（8bit）を定義する。サイズは 10 進数（根 10）として表現する。したがって、“0”..“9”の数字で表現され、単位は bytes である。（Mbytes, GB 等ではない）

型: 任意文字列（最低限保証すべき最大値: 30 文字）

繰り返し: <size>要素は, <technical>要素中に 0 回または 1 回出現する .

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<technical>  
  <size>568</size>  
</technical>
```

3.1.3.1.4.3 <location>要素

説明: <location>要素は, この学習オブジェクトにアクセスするために使用する文字列を定義する . Universal Resource Locator (URL) 等の格納場所や, Universal Resource Identifier (URI) のような格納場所を探す方法である . このリストの最初の要素は, 望ましい格納場所とする .

型: 任意文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <location>要素は, <technical>要素中に 1 回以上出現する . 最低限保証すべき最大値は 10 項目である .

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	1 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<technical>  
  <location>http://host/id</location>
```

</technical>

3.1.3.1.4.4 <requirement>要素

説明: <requirement>要素は、この学習オブジェクトを使用する場合の技術的な要件を定義する。複数の必要条件がある場合は、すべてが要求される。つまり、論理接続詞は AND である。

型: 親要素

繰り返し: <requirement>要素は、<technical>要素中に 0 回以上出現する。最低限保証すべき最大値は 40 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <orComposite>

例:

```
<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>
```

3.1.3.1.4.4.1. <orComposite>要素

説明: <orComposite>要素は、多数の必要条件を複合したものを記述する。複数の必要条件は、構成要素の必要条件のうち 1 つが満たされるということである。つまり、論理接続詞は OR である。

型: 親要素

繰り返し: <orComposite>要素は<requirement>要素中に 0 回以上出現する。最低限保証すべき最大値は 40 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

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

例:

```
<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>
```

3.1.3.1.4.4.1.1. <type>要素

説明: <type>要素は、この学習オブジェクトを使用するのに要求される、ハードウェア・ソフトウェア・ネットワーク等の技術を記述する。

型: ポキャブラリ

繰り返し: <type>要素は、<orComposite>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1

アセット	0または1
------	-------

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- OS (operating system)
- ブラウザ (browser)

例:

```

<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>

```

3.1.3.1.4.4.1.2. <name>要素

説明: <name>要素は、この学習オブジェクトを使用するのに要求される技術の名称を記述する。

型: ボキャブラリ

繰り返し: <name>要素は、<orComposite>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0または1
アクティビティ	0または1
SCO	0または1
SCA	0または1
アセット	0または1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

If <value> child element of <type> element = 'operating system'

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

If <value> child element of <type> element = 'browser'

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

例:

```
<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>
```

3.1.3.1.4.4.1.3. <minimumVersion>要素

説明: <minimumVersion>要素は、この学習オブジェクトを使用する際に要求される技術の最低限可能なバージョンを表す。

型: 任意文字列 (最低限保証すべき最大値: 30 文字)

繰り返し: <minimumVersion>要素は、<orComposite>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1

SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<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>
```

3.1.3.1.4.4.1.4. <maximumVersion>要素

説明: <maximumVersion>要素は、この学習オブジェクトを使用する際に要求される技術の最上限可能なバージョンを表す。

型: 任意文字列（最低限保証すべき最大値: 30 文字）

繰り返し: <maximumVersion>要素は、<orComposite>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<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>
</requirement>
</technical>

```

3.1.3.1.4.5 <installationRemarks>要素

説明: <installationRemarks>要素は、この学習オブジェクトのインストール方法を提供する。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <installationRemarks>要素は、<technical>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```

<technical>
  <installationRemarks>
    <string language="en">Installation remarks placed here</string>
  </installationRemarks>
</technical>

```

3.1.3.1.4.6 <otherPlatformRequirements>要素

説明: <otherPlatformRequirements>要素は、その他のソフトウェア・ハードウェアについての要求事項を記述する。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <otherPlatformRequirements>要素は, <technical>要素中に 0 回または 1 回出現する.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<technical>
  <otherPlatformRequirements>
    <string language="en">Other platform requirements placed here</string>
  </otherPlatformRequirements>
</technical>
```

3.1.3.1.4.7 <duration>要素

説明: <duration>要素は, 学習オブジェクトを想定される速さで実行するときの継続時間を記述する.

型: 期間 (Duration)

繰り返し: <duration>要素は, <technical>要素中に 0 回または 1 回出現する.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 期間形式 (Duration Type) を参照

例:

```
<technical>
  <duration>
    <duration>P5Y</duration>
    <description>
      <string language="en">Length of time to play the simulation</string>
    </description>
  </duration>
</technical>
```

3.1.3.1.5 <educational>要素

説明: <educational>要素は、学習オブジェクトのキーとなる教育的または教育的特性を記述する。

型: 親要素

繰り返し: <educational>要素は、トップレベルの<lom>要素中に 0 回または 1 回出現する。最低限保証すべき最大値は 100 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

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

3.1.3.1.5.1 <interactivityType>要素

説明: <interactivityType>要素は、この学習オブジェクトでサポートされたが学習の顕著な形態を定義する。

型: ボキャブラリ

繰り返し: <interactivityType>要素は, <educational>要素中に 0 回または 1 回出現する.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 対話型 (active)
- 提示型 (expositive)
- 対話 / 提示混合型 (mixed)

例:

```
<educational>
  <interactivityType>
    <source>LOMv1.0</source>
    <value>active</value>
  </interactivityType>
</educational>
```

3.1.3.1.5.2 <learningResourceType>要素

説明: <learningResourceType>要素は, 学習オブジェクトの種類を特定する. もっとも重要な種類を最初に記述する.

型: ボキャブラリ

繰り返し: <learningResourceType>要素は, <educational>要素中に 0 回以上出現する. 最低限保証すべき最大値は 10 項目である.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- ドリル (exercise)
- シミュレーション (simulation)
- 問題集 (questionnaire)
- 図形 (diagram)
- 画像 (figure)
- グラフ (graph)
- インデックス (index)
- スライド (slide)
- 表 (table)
- 説明文 (narrative text)
- 試験 (exam)
- 実験 (experiment)
- 問題記述 (problem statement)
- 自己評価 (self assessment)
- 講義 (lecture)

例:

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

3.1.3.1.5.3 <interactivityLevel>要素

説明: <interactivityLevel>要素は、この学習オブジェクトを特徴付ける相互作用の程度を定義する。このコンテキストの相互作用は、学習オブジェクトの様相や動作に、学習者が影響を及ぼすことができる程度を参照する。

型: ボキャブラリ

繰り返し: <interactivityLevel>要素は、<educational>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1

SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されたボキャブラリ (<source>要素は LOMv1.0 を設定)

- 大変低い (very low)
- 低い (low)
- 普通 (medium)
- 高い (high)
- 大変高い (very high)

例:

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

3.1.3.1.5.4 <semanticDensity>要素

説明: <semanticDensity>要素は、学習オブジェクトの有効性の程度を定義する。学習オブジェクトの意味的な密度は、そのサイズや範囲（オーディオやビデオのように限られた時間のものは）、継続時間などで判断される。学習オブジェクトの意味的な密度は、その難易度には依存しない。提示型の素材はよい例であるが、対話型の資源でも同様に使用される。

型: ボキャブラリ

繰り返し: <semanticDensity>要素は、<educational>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 大変低い (very low)
- 低い (low)
- 普通 (medium)
- 高い (high)
- 大変高い (very high)

例:

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

3.1.3.1.5.5 <intendedEndUserRole>要素

説明: <intendedEndUserRole>要素は、この学習オブジェクトの設計時に想定されている主要な利用者 (複数) を定義するために用いられる。もっとも関係の深いものを最初に記述する。

型: ボキャブラリ

繰り返し: <intendedEndUserRole>要素は、<educational>要素中に 0 回または 1 回出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 教師 (teacher)
- 製作者 (author)
- 学習者 (learner)
- 管理者 (manager)

例:

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

3.1.3.1.5.6 <context>要素

説明: <context>要素は、学習や、この学習オブジェクトの利用が想定される典型的な環境を定義する。

型: ボキャブラリ

繰り返し: <context>要素は、<educational>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されたボキャブラリ (<source>要素は LOMv1.0 を設定)

- 学校 (school)
- 高等教育 (higher education)
- 訓練 (training)
- その他 (other)

例:

```
<educational>
  <context>
    <source>LOMv1.0</source>
    <value>higher education</value>
  </context>
</educational>
```

3.1.3.1.5.7 <typicalAgeRange>要素

説明: <typicalAgeRange>要素は、想定されるユーザの年齢層を定義するために用いられる。この年齢は発達年齢 (developmental age) であり、年齢 (chronological age) ではない。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <typicalAgeRange>要素は、<educational>要素中に 0 回以上出現する。最低限保証すべき最大値は 5 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<educational>
  <typicalAgeRange>
    <string language="en">adult pilot with 3 years erience</string>
  </typicalAgeRange>
</educational>
```

3.1.3.1.5.8 <difficulty>要素

説明: <difficulty>要素は、この学習オブジェクトを標準的な対象ユーザが実行し通過するのがどのくらい難しいかを定義するために用いられる。

型: ボキャブラリ

繰り返し: <difficulty>要素は<educational>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されたボキャブラリ (<source>要素は LOMv1.0 を設定)

- 大変易しい (very easy)
- 易しい (easy)
- 標準レベル (medium)
- 難しい (difficult)
- 大変難しい (very difficult)

例:

```
<educational>
  <difficulty>
    <source>LOMv1.0</source>
    <value>medium</value>
  </difficulty>
</educational>
```

3.1.3.1.5.9 <typicalLearningTime>要素

説明: <typicalLearningTime>要素は、この学習オブジェクトを標準的な対象ユーザが実行し通過するのにかかる大体の、または標準的な時間を定義するために用いられる。

型: 期間

繰り返し: <typicalLearningTime>要素は<educational>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 期間形式 (Duration Type) を参照

例:

```
<educational>
  <typicalLearningTime>
    <duration> PT1H30M </duration>
    <description>
      <string language="en">It takes this long to complete the
course</string>
    </description>
  </typicalLearningTime>
</educational>
```

3.1.3.1.5.10 <description>要素

説明: <description>要素は、学習資源の使い方に関するコメントを記述する。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <description>要素は、<educational>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<educational>
  <description>
    <string language="en">Used for training on in-flight refueling</string>
  </description>
</educational>
```

3.1.3.1.5.11 <language>要素

説明: この<language>要素は、この学習オブジェクトの標準的な想定ユーザが使用する言語を記述する。

型: 任意文字列 (最低限保証すべき最大値: 100 文字)

繰り返し: <language>要素は、<educational>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0回以上
アクティビティ	0回以上
SCO	0回以上
SCA	0回以上
アセット	0回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<educational>
  <language>en</language>
</educational>
```

3.1.3.1.6 <rights> 要素

説明: <rights>要素は、学習オブジェクトの著作権や使用条件について記述する。

型: 親要素

繰り返し: <rights>要素はトップレベルの<lom>要素中に 1 回だけ出現しなければならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1回のみ
アクティビティ	1回のみ
SCO	1回のみ
SCA	1回のみ
アセット	1回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

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

3.1.3.1.6.1 <cost>要素

説明: <cost>要素は、この学習オブジェクトを使用するために使用料が求められるかどうかを定義する。

型: ボキャブラリ

繰り返し: <cost>要素は, <rights>要素中に 1 回だけ出現しなければならない.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- yes
- no

例:

```
<rights>
  <cost>
    <source>LOMv1.0</source>
    <value>no</value>
  </cost>
</rights>
```

3.1.3.1.6.2 <copyrightAndOtherRestrictions>要素

説明: <copyrightAndOtherRestrictions>要素は, 著作権や他の制約がこの学習オブジェクトの使用に適用されるかを取り込む.

型: ボキャブラリ

繰り返し: <copyrightAndOtherRestrictions>要素は, <rights>要素中に 1 回だけ書
告げなければならない.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	1 回のみ

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- yes
- no

例:

```
<rights>
  <copyrightAndOtherRestrictions>
    <source>LOMv1.0</source>
    <value>>no</value>
  </copyrightAndOtherRestrictions>
</rights>
```

3.1.3.1.6.3 <description>要素

説明: <description>要素は, 学習資源の使用条件についてのコメントを含む.

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <description>要素は, <rights>要素中に 0 回または 1 回出現する.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<rights>
  <description>
    <string language="en">You are free to use this object any way you
like</string>
  </description>
</rights>
```

3.1.3.1.7 <relation>要素

説明: <relation>要素は, この学習オブジェクトと他の学習オブジェクトの関係を定義している. 複数の関係を定義する場合は, 複数のインスタンスを持たせることができる. もし, 対象の学習オブジェクトが複数の場合は, 書く対象について新しく関係付けのインスタンスを定義する.

型: 親要素

繰り返し: <relation>要素は, トップレベルの<lom>要素中に 0 回以上出現する. 最低限保証すべき最大値は 100 項目である.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <kind>
- <resource>

3.1.3.1.7.1 <kind>要素

説明: <kind>要素は, この学習オブジェクトと 3.1.3.1.7.2 <resource>によって識別される対象の学習オブジェクトとの関係を表す.

型: ボキャブラリ

繰り返し: <kind>要素は, <relation>要素中に 0 回または 1 回出現する.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 包含元 (ispartof)
- 包含 (haspart)
- バージョン元 (isversionof)
- 一つのバージョン (hasversion)
- 形式元 (isformatof)
- 一つの形式 (hasformat)
- 参照先 (references)
- 参照元 (isreferencedby)
- 基礎 (isbasedon)
- 応用 (isbasisfor)
- 必須 (requires)
- 要求元 (isrequiredby)

例:

```
<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>requires</value>
  </kind>
  <resource>
    <identifier>
      <catalog>Sample Catalog</catalog>
      <entry>Sample Entry</entry>
    </identifier>
    <description>
      <string language="en">Description of resource</string>
    </description>
  </resource>
</relation>
```

3.1.3.1.7.2 <resource>要素

説明: <resource>要素は、この関係の対象となる学習オブジェクトを表す。

型: 親要素

繰り返し: <resource>要素は<relation>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- <identifier>
- <description>

例:

```
<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>requires</value>
  </kind>
  <resource>
    <identifier>
      <catalog>Sample Catalog</catalog>
      <entry>Sample Entry</entry>
    </identifier>
    <description>
      <string language="en">Description of resource</string>
    </description>
  </resource>
</relation>
```

3.1.3.1.7.2.1. <identifier>要素

説明: <identifier>要素は、対象となる学習オブジェクトを識別する、世界的にユニークな識別子を定義する。

型: 親要素

繰り返し: <identifier>要素は、<metaMetadata>要素中に 0 回以上出現しなければならない。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <catalog>
- <entry>

例:

```
<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>requires</value>
  </kind>
  <resource>
    <identifier>
      <catalog>Sample Catalog</catalog>
      <entry>Sample Entry</entry>
    </identifier>
    <description>
      <string language="en">Description of resource</string>
    </description>
  </resource>
</relation>
```

3.1.3.1.7.2.1.1. <catalog>要素

説明: <catalog>要素は、このエントリの識別空間またはカタログスキーマの名前もしくは指定子を記述する。

型: 任意文字列（最低限保証すべき最大値: 1000 文字）

繰り返し: <catalog>要素は、<identifier>要素中に 0 回または 1 回出現しなければならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>requires</value>
  </kind>
  <resource>
    <identifier>
      <catalog>Sample Catalog</catalog>
      <entry>Sample Entry</entry>
    </identifier>
    <description>
      <string language="en">Description of resource</string>
    </description>
  </resource>
</relation>
```

3.1.3.1.7.2.1.2. <entry>要素

説明: <entry>要素は、識別空間もしくはカタログ化スキーマ内の識別子の値を定義する。

型: 任意文字列（最低限保証すべき最大値: 1000 文字）

繰り返し: <entry>要素は、<identifier>要素中に 0 回または 1 回出現しなければならない。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>requires</value>
  </kind>
  <resource>
    <identifier>
      <catalog>Sample Catalog</catalog>
      <entry>Sample Entry</entry>
    </identifier>
    <description>
      <string language="en">Description of resource</string>
    </description>
  </resource>
</relation>
```

3.1.3.1.7.2.2. <description>要素

説明: <description>要素は、対象となる学習オブジェクトの記述を含む。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <description>要素は、<resource>要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>requires</value>
  </kind>
  <resource>
    <identifier>
      <catalog>Sample Catalog</catalog>
      <entry>Sample Entry</entry>
    </identifier>
    <description>
      <string language="en">Description of resource</string>
    </description>
  </resource>
</relation>
```

3.1.3.1.8 <annotation>要素

説明: <annotation>要素は、この学習オブジェクトの教育的使用におけるコメントや、コメントがいつ誰によって作成されたかという情報を供給する。このカテゴリは、教育者に学習オブジェクトの評価や使用のための提案を共有させる。

型: 親要素

繰り返し: <annotation>要素は、トップレベルの <lom>要素中に 0 回以上出現する。最低限保証すべき最大値は 30 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <entity>
- <date>
- <description>

3.1.3.1.8.1 <entity>要素

説明: <entity>要素は、この注釈を作成したもの（人物や組織）を含む。

型: VCard（最低限保証すべき最大値: 1000 文字）

繰り返し: <entity>要素は、<annotation>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0または1
アクティビティ	0または1
SCO	0または1
SCA	0または1
アセット	0または1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: VCard (VCard Type) を参照

例:

```
<annotation>
  <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Adminstrator\,Assistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</
entity>
</annotation>
```

3.1.3.1.8.2 <date>要素

説明: <date>要素は、この注釈を作成した日付を含む。

型: 日付形式

繰り返し: <date>要素は、<annotation>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0または1
アクティビティ	0または1
SCO	0または1
SCA	0または1
アセット	0または1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 日付形式 (DateTime Type) を参照

例:

```
<annotation>
  <date>
    <dateTime>2001-07-30T10:14</dateTime>
    <description>
      <string language="en">description goes here</string>
    </description>
  </date>
</annotation>
```

3.1.3.1.8.3 <description>要素

説明: <description>要素は、この注釈の内容を含む。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <description>要素は、<annotation>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<annotation>
  <description>
    <string language="en">This simulation can be used in conjunction with
the in-flight refueling course</string>
  </description>
</annotation>
```

3.1.3.1.9 <classification>要素

説明: <classification>要素は、特定の分類規則 (システム) でどこにこの学習オブジェクトが属するかを記述する。複数の分類を定義する場合は、このカテゴリに複数のインスタンスを置くことができる。

型: 親要素

繰り返し: <classification>要素は、トップレベルの<lom>要素中に 0 回以上出現する。最低限保証すべき最大値は 40 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <purpose>
- <taxonPath>
- <description>
- <keyword>

3.1.3.1.9.1 <purpose>要素

説明: <purpose>要素は, この学習オブジェクトの分類目的を定義する .

型: ボキャブラリ

繰り返し: <purpose>要素は, <classification>要素中に 0 回または 1 回出現する .

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ
SCA	1 回のみ
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: ボキャブラリ (Vocabulary Type) を参照

LOM に定義されるボキャブラリ (<source>要素は LOMv1.0 を設定)

- 教科 (discipline)
- アイデア (idea)
- 必要条件 (prerequisite)
- 学習目的 (educational objective)
- アクセス可能性 (accessibility)
- 制限 (restrictions)
- 教育的レベル (educational level)
- スキルレベル (skill level)
- セキュリティレベル (security level)
- コンピテンシ (competency)

例:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>educational objective</value>
  </purpose>
</classification>
```

3.1.3.1.9.2 <taxonPath>要素

説明: <taxonPath>要素は、特定の分類規則（システム）における分類体系パス（＝タクソノミ）を定義する。後ろに続くレベルはそれぞれ、前のレベルにおける細かい定義である。同じまたは異なるパスが同じ特性を記述する場合もありうる。

型: 親要素

繰り返し: <taxonPath>要素は、トップレベルの<classification>要素中に 0 回以上出現する。最低限保証すべき最大値は 15 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <source>
- <taxon>

例:

```
<classification>
  <taxonPath>
    <source>
      <string language="en">DDC</string>
    <source>
      <taxon>
        <id>342.12.1</id>
        <entry>simulation</entry>
      </taxon>
    </taxonPath>
  </classification>
```

3.1.3.1.9.2.1. <source>要素

説明: <source>要素は、分類体系（システム）の名前を定義する。このデータ要素は、“公式”タクソノミまたはユーザが定義したタクソノミの識別に使用される。

型: 言語文字列（最低限保証すべき最大値: 1000 文字）

繰り返し: <source>要素は、<taxonpath>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
----------------------------	------

コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<classification>
  <taxonPath>
    <source>
      <string language="en">DDC</string>
    <source>
    <taxon>
      <id>342.12.1</id>
      <entry>simulation</entry>
    </taxon>
  </taxonPath>
</classification>
```

3.1.3.1.9.2.2. <taxon>要素

説明: <taxon>要素は、タクソノミで用いられる特定の用語である。各分類項目 (taxon) はラベルや用語の定義されたノードである。各分類項目 (taxon) はまた、標準化された参照のための英数字の名称、あるいは識別子を持つ。ラベルやエントリは、特定の各分類項目 (taxon) を示すために使用される。各分類項目 (taxon) の順番付きリスト (ordered list) は、分類体系へのパス (taxonomic stairway) を生成する。これは、一般的なエントリから特徴的なエントリへのパスである。

型: 親要素

繰り返し: <taxon>要素は、<taxonPath>要素中に 0 回以上出現する。最低限保証すべき最大値は 15 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 回以上
アクティビティ	0 回以上
SCO	0 回以上
SCA	0 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- <id>
- <entry>

例:

```
<classification>
  <taxonPath>
    <source>
      <string language="en">DDC</string>
    <source>
      <taxon>
        <id>342.12.1</id>
        <entry>simulation</entry>
      </taxon>
    </taxonPath>
  </classification>
```

3.1.3.1.9.2.2.1. <id>要素

説明: <id>要素は、タクソノミのソースによって与えられる、数字や文字の組み合わせである各分類項目の識別子である。

型: 任意文字列 (最低限保証すべき最大値: 100 文字)

繰り返し: <id>要素は、<taxon>要素中に 0 回または 1 回出現する。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<classification>
  <taxonPath>
    <source>
      <string language="en">DDC</string>
    <source>
      <taxon>
        <id>342.12.1</id>
        <entry>simulation</entry>
      </taxon>
    </taxonPath>
  </classification>
```

3.1.3.1.9.2.2.2. <entry>要素

説明: <entry>要素は, 各分類項目のテキストラベルを表す.

型: 言語文字列 (最低限保証すべき最大値: 500 文字)

繰り返し: <entry>要素は, <taxon>要素中に 0 回または 1 回出現する.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	0 または 1
アクティビティ	0 または 1
SCO	0 または 1
SCA	0 または 1
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- なし

例:

```
<classification>
  <taxonPath>
    <source>
      <string language="en">DDC</string>
    <source>
      <taxon>
        <id>342.12.1</id>
        <entry>simulation</entry>
      </taxon>
    </taxonPath>
  </classification>
```

3.1.3.1.9.3 <description>要素

説明: <description>要素は, この特定の分類体系における 3.1.3.1.9.1:Purpose の分類項目の指定 (教科, アイデア, スキルレベル, 学習目的など) と学習オブジェクトの関係についての記述を含む.

型: 言語文字列 (最低限保証すべき最大値: 2000 文字)

繰り返し: <description>要素は, <classification>要素中に 0 回または 1 回出現する.

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回のみ
アクティビティ	1 回のみ
SCO	1 回のみ

SCA	1 回のみ
アセット	0 または 1

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<classification>
  <description>
    <string language="en">simulation object</langstring>
  </description>
</classification>
```

3.1.3.1.9.4 <keyword>要素

説明: <keyword>要素は、この特定の分類体系における 3.1.3.1.9.1:Purpose の分類項目の指定 (アクセス可能性, セキュリティレベルなど) と学習オブジェクトの関係を表すキーワードやフレーズを含む。もっとも関係の深いものを最初に記述する。

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <keyword>要素は、<classification>要素中に 0 回以上出現する。最低限保証すべき最大値は、40 項目である。

メタデータアプリケーション プロファイルタイプ	繰り返し
コンテンツアグリゲーション	1 回以上
アクティビティ	1 回以上
SCO	1 回以上
SCA	1 回以上
アセット	0 回以上

SCORM アプリケーションプロファイル要求事項

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<classification>
  <keyword>
    <string language="en">cockpit</string>
  </keyword>
  <keyword>
    <string language="en">F18</string>
  </keyword>
</classification>
```

3.1.3.2 共通要素型

メタデータには、繰り返し使われる共通の構造がある。これらの共通な構造や要素の使用は、実装において共通のデータ構造を促すであろう。共通の構造や要素の詳細を以下に示す。

3.1.3.2.1 任意文字列 (CharacterString Type)

説明: 任意文字列は、単に xs:string 型の別名であり、ただ文字列だけを表すものである。

3.1.3.2.2 言語文字列 (LangString Type)

説明: 言語文字列は、人間の言語でフレーズを記述するために用いられる。

要素:

- <string>

例:

```
<general>
  <keyword>
    <string language="en">metadata</string>
    <string language="nl">metadata</string>
    <string language="fr">metadonnees</string>
  </keyword>
</general>
```

3.1.3.2.2.1 <string>要素

説明: <string>要素は、人間の言語での実際のフレーズを含んでいる。

型: 任意文字列 (最低限保証すべき最大値は親要素に依存する)

繰り返し: <string>要素は、その親要素中に 0 回以上出現する。最低限保証すべき最大値は 10 項目である。

属性:

- language – 要素内容の言語を表す。language 属性は 100 文字の最低限保証すべき最大値の文字列として表現される。language 属性はオプションである。<metaMetadata>要素の<language>要素は、すべての言語文字列値の既定の言語を表す。もし language 属性が個々の<string>要素に存在しない場合、string で保持された値は<metaMetadata>要素の<language>要素で定義された言語を表す。

子要素:

- なし

例:

```
<general>
  <keyword>
    <string language="en">metadata</string>
    <string language="nl">metadata</string>
    <string language="fr">metadonnees</string>
  </keyword>
</general>
```

3.1.3.2.3 日付形式 (DateTime Type)

説明: 日付形式は、時を記述するために用いられる。

要素:

- <dateTime>
- <description>

例:

```
<annotation>
  <date>
    <dateTime> 2001-07-30T10:14</dateTime>
  </date>
  <description>
    <string language="en">description goes here</string>
  </description>
</annotation>
```

3.1.3.2.3.1 <dateTime> 要素

説明: <dateTime>要素は、少なくとも 1 秒の精度を持つ時を表す。

型: 任意文字列 (最低限保証すべき最大値: 200 文字)

書式:

YYYY[-MM][-DD[Thh[:mm[:ss[:s[TZD]]]]]]

位置:

- YYYY = 年 4 桁 (>=0001)
- MM = 月 2 桁 (01 ~ 12)
- DD = 日 2 桁 (01 ~ 31)
- hh = 時 2 桁 (00 ~ 23)
- mm = 分 2 桁 (00 ~ 59)
- ss = 秒 2 桁 (00 ~ 59)
- s = 秒の小数を 1 桁かそれ以上で表現する 10 進数
- TZD = 時間帯指定 (time zone designator)
- 少なくとも, 年は 4 桁存在しなければならない . 日付形式の追加の部分が含まれる場合, 文字列リテラル“-“, “T“, “:“, “.”は日付形式 (datetime) の一部である .

繰り返し: <datetime>要素は, その親要素中に 0 回または 1 回出現する .

子要素:

- なし

例:

```
<annotation>
  <date>
    <dateTime>2001-07-30T10:14</dateTime>
    <description>
      <string language="en">description goes here</string>
    </description>
  </date>
</annotation>
```

3.1.3.2.3.2 <description>要素

説明: <description>要素は, 日付の説明を含む .

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <description>要素は, その親要素中に 0 回または 1 回出現する .

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (LangString Type) を参照

例:

```
<annotation>
  <date>
    <dateTime>2001-07-30T10:14</dateTime>
    <description>
      <string language="en">description goes here</string>
    </description>
  </date>
</annotation>
```

3.1.3.2.4 期間形式 (Duration Type)

説明: 期間形式は、時間の間隔を記述するために用いられる。

要素:

- <duration>
- <description>

例:

```
<educational>
  <typicalLearningTime>
    <duration>PT1H30M</duration>
    <description>
      <string language="en">It takes this long to complete the
course</string>
    </description>
  </typicalLearningTime>
</educational>
```

3.1.3.2.4.1 <duration>要素

説明: <duration>要素は、少なくとも 1 秒の精度を持つ時間の間隔を表す。

型: 任意文字列 (最低限保証すべき最大値: 200 文字)

書式:

P[yY][mM][dD][T[hH][nM][s[.s]S]]

位置:

- y = 年
- m = 月
- d = 日
- h = 時
- n = 分
- s = 秒または秒の小数
- 文字列リテラル“P”, “Y”, “M”, “D”, “T”, “H”, “M”, “S”は, それぞれに対応する 0 以外の値が存在する場合, 表記しなければならない.

繰り返し: <duration>要素は, その親要素中に 0 回または 1 回出現する.

子要素:

- なし

例:

```
<educational>
  <typicalLearningTime>
    <duration>PT1H30M</duration>
    <description>
      <string language="en">It takes this long to complete the
course</string>
    </description>
  </typicalLearningTime>
</educational>
```

3.1.3.2.4.2 <description>要素

説明: <description>要素は, 期間の説明を含む.

型: 言語文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <description>要素は, その親要素中に 0 回または 1 回出現する.

子要素:

- 3.1.3.2 章 – 共通要素型: 言語文字列 (*LangString Type*) を参照

例:

```
<educational>
  <typicalLearningTime>
    <duration>PT1H30M</duration>
    <description>
      <string language="en">It takes this long to complete the
course</string>
    </description>
  </typicalLearningTime>
</educational>
```

3.1.3.2.5 ボキャブラリ (Vocabulary Type)

説明: ボキャブラリは , ボキャブラリ構造を記述するために用いられる .

要素:

- <source>
- <value>

例:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>educational objective</value>
  </purpose>
</classification>
```

3.1.3.2.5.1 <source>要素

説明: <source>要素は , “LOMv1.0” , もしくは URI のような値の出典の表示を含む .

型: 任意文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <source>要素は , その親要素中に 0 回または 1 回出現する .

子要素:

- なし

例:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>educational objective</value>
  </purpose>
</classification>
```

3.1.3.2.5.2 <value>要素

説明: <value>要素は、実際の値を含む。出典が“LOMv1.0”である場合、値はその親要素で定義されたリストによるものとする。

型: 任意文字列 (最低限保証すべき最大値: 1000 文字)

繰り返し: <value>要素は、その親要素中に 0 回または 1 回出現する。

子要素:

- なし

例:

```
<classification>
  <purpose>
    <source>LOMv1.0</source>
    <value>educational objective</value>
  </purpose>
</classification>
```

3.1.3.2.6 VCard (VCard Type)

説明: VCard は、個人や組織などの構成要素を記述するために用いられる。VCard は、伝統的な名刺によくみられるパーソナルな情報の交換を自動化する。VCard 仕様は、「仮想的な」名刺を定義する。VCard は、名前や住所、電話番号、E メールアドレス等の情報を格納できる。

要素:

- なし

例:

```
<annotation>
  <entity>BEGIN:VCARD\nFN:Joe Friday\nTEL:+1-919-555-7878\nTITLE:Area
Adminstrator\,Assistant\nEMAIL\;TYPE=INTERN\nET:jfriday@host.com\nEND:VCARD\n</entity>
</annotation>
```

3.1.3.3 XML 拡張のメカニズム

IEEE LOM 仕様は、IEEE LOM メタデータインスタンスを通して、コミュニティが独自に命名したメタデータ要素を設定できるようにしている。バインディングで親要素として類別されたすべての要素は、拡張できるように考慮されている。SCORM はこの実践を進めているが、相互運用性の観点からこの実践に警告を発する。将来、IEEE LOM 情報モデルを拡張することが ADL に要求されるかもしれない。もし SCORM バージョン 1.3 メタデータインスタンスがコミュニティ固有の拡張を含んでいても、無効とはみなされないだろう。このようなメタ

データインスタンスを取り扱う準拠システムは、SCORM バージョン 1.3 で定義されるスキーマでない XML 要素を無視することになる。ADL コミュニティは、メタデータ情報モデルへのあらゆる潜在的な追加項目を供給することを要求されている。ADL 技術チームは、特定の仕様 / 標準グループを通して、追加作業を続けていくであろう。

3.1.3.3.1 ボキャブラリ拡張のメカニズム

IEEE LOM 仕様はまた、ボキャブラリ型で定義された要素について、コミュニティが独自に定義したボキャブラリの値を設定できるようにしている。もし SCORM バージョン 1.3 メタデータインスタンスがコミュニティ固有のボキャブラリ拡張を含んでいても、無効とはみなされないだろう。注意すべきは、ボキャブラリ拡張を使用することが、相互運用性を妨げるかもしれないということだろう。“LOMv1.0”の値がボキャブラリ<source>要素中にある場合、<value>要素が情報モデルからリストされた値を含まなければならないことに留意せよ。

3.1.4. SCORM メタデータアプリケーションプロファイル

SCORM メタデータアプリケーションプロファイルは、IEEE LTSC LOM 仕様の ADL 環境への統合を記述している。アプリケーションプロファイルは、さらにメタデータのタイプと、2.1 節で説明したコンテンツアグリゲーションモデルへのその適用方法を記述している。SCORM は、コンテンツアグリゲーション、SCO、SCA、アクティビティ、アセットの 5 タイプの学習コンテンツメタデータを識別する。以下の節では、これらのメタデータアプリケーションプロファイルについて説明する。

SCORM において、SCORM メタデータアプリケーションプロファイルは、IEEE LTSC LOM 仕様を特殊化したものである。SCORM は、仕様の適用において追加の制約を課している。この特殊化により SCORM メタデータアプリケーションプロファイルは、メタデータの使用をすでに定めている IMS コンテンツパッケージング仕様の全体によくフィットする。

SCORM では、2 つの主なメタデータタイプが存在する。文脈依存と文脈非依存である。文脈依存メタデータは、ある特定の文脈 - 通常は、コンテンツより大きなユニット内での「学習」の流れ - で学習コンテンツを記述する。文脈非依存メタデータは、ある文脈（または利用方法）に依存しない様に学習コンテンツを説明するメタデータである。この 2 つのメタデータの区別は、2.3 節のコンテンツパッケージングでさらに述べる。

3.1.4.1 パッケージレベルメタデータ

パッケージレベルメタデータは、パッケージ全体を記述するためのメタデータである。SCORM はこれまで、パッケージレベルのメタデータについての要求事項が無かった。ただ一つの要求事項は、そのメタデータが IEEE XML バインデ

ィングに準拠しているかということだけである。すべての要素はオプションとして定義されている。

3.1.4.2 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、コンテンツリポジトリからの発見と再利用を容易にし、コンテンツアグリゲーションについての情報記述を供給するために使われる。コンテンツアグリゲーションメタデータは、なんのためのものか、誰が使用できるのか、誰がコントロールするのか等の情報と、コンテンツアグリゲーション名、説明記述、バージョン等、外部から検索可能な情報である。

3.1.4.3 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アクティビティに適用できるメタデータであり、特定のコンテキストに依存しないアクティビティの属性や特徴についての記述情報を供給する。このメタデータは、このようなアクティビティを、例えばコンテンツリポジトリからの発見やそれを再利用するといったことを容易にする。

3.1.4.4 SCO メタデータ

SCO メタデータは、SCO に適用できるメタデータであり、特定のコンテキストに依存しない学習資源の属性や特徴についての記述情報を供給する。このメタデータは、例えばコンテンツリポジトリから、学習資源の発見と再利用を容易にするために使われる。SCO メタデータは、SCO の属性や特徴を記述するメタデータであり、また特定のコンテンツアグリゲーション構造に関連しないメタデータ（文脈非依存メタデータ）であり、コンテンツタイトル、説明記述、制作年月日、バージョン等の、外部から検索可能な情報である。

3.1.4.5 SCA メタデータ

SCA メタデータは、SCA に適用できるメタデータであり、特定のコンテキストに依存しない学習資源の属性や特徴についての記述情報を供給する。このメタデータは、このような学習資源を、例えばコンテンツリポジトリからの発見やそれを再利用するといったことを容易にする。SCA メタデータは、SCA の属性や特徴を記述するメタデータであり、また特定のコンテンツアグリゲーション構造に関連しないメタデータ（文脈非依存メタデータ）であり、コンテンツタイトル、説明記述、制作年月日、バージョン等の、外部から検索可能な情報である。

3.1.4.6 アセットメタデータ

アセットメタデータは、挿し絵、ドキュメント、メディアストリーム等のアセットに適用可能なメタデータであり、学習コンテンツに依存しないアセットの属性や特徴についての記述情報を供給する。このメタデータは、主に学習コンテンツの生成をするときに、例えばコンテンツリポジトリからアセットを発見し再利用を容易にするために使われる。アセットメタデータは、特定の文脈に依存しない方法でアセットの属性や特徴を記述するメタデータであり、またアセットのタイトルや記述説明、制作年月日、バージョンのような外部から検索可能な情報、さらに共有可能なアセットの検索可能なリポジトリの生成に使用可能な情報である。

3.1.4.7 SCORM メタデータアプリケーションプロファイルの要求条件

次の表 2.2.4.4a は、上記の各メタデータアプリケーションプロファイルへの要求条件を定義する。各メタデータアプリケーションプロファイルは、各メタデータ要素への対応する要求条件と共に列挙されている。これらの要求条件は、必ずしもすべてのコンテンツアグリゲーション、SCO、SCA、アクティビティ、アセットの属性や特徴をメタデータによって記述する必要があるという意味ではないことに注意せよ。しかしながらこの要求は、メタデータがコンテンツアグリゲーションやSCO、SCA、アクティビティ、アセットの記述に使われるときにはいつでも適用される。“M”はその要素が必須であることを示す。“O”は、その要素がオプション（選択可能）であることを示す。

名前	パッケージ	コンテンツアグリゲーション	アクティビティ	SCO	SCA	アセット
1.0 一般(General)	O	M	M	M	M	M
1.1 識別子(Identifier)	O	M	M	M	M	M
1.1.1 カタログ(Catalog)	O	M	M	M	M	M
1.1.2 エントリ(Entry)	O	M	M	M	M	M
1.2 タイトル(Title)	O	M	M	M	M	M
1.3 言語(Language)	O	O	O	O	O	O
1.4 記述(Description)	O	M	M	M	M	M
1.5 キーワード(Keyword)	O	M	M	M	M	O
1.6 対象範囲(Coverage)	O	O	O	O	O	O
1.7 基本構造(Structure)	O	O	O	O	O	O
1.8 アグリゲーションレベル(Aggregation Level)	O	O	O	O	O	O
2.0 ライフサイクル(Life Cycle)	O	M	M	M	M	O
2.1 バージョン(Version)	O	M	M	M	M	O
2.2 状態(Status)	O	M	M	M	M	O
2.3 コントリビュート(Contribute)	O	O	O	O	O	O
2.3.1 役割(Role)	O	O	O	O	O	O
2.3.2 情報(Entity)	O	O	O	O	O	O
2.3.3 日付(Date)	O	O	O	O	O	O
3.0 メタメタデータ(Meta-Metadata)	O	M	M	M	M	M

3.1 識別子(Identifier)	O	M	M	M	M	M
3.1.1 カタログ(Catalog)	O	M	M	M	M	M
3.1.2 エントリ(Entry)	O	M	M	M	M	M
3.2 コントリビュート (Contribute)	O	O	O	O	O	O
3.2.1 役割(Role)	O	O	O	O	O	O
3.2.2 情報(Entity)	O	O	O	O	O	O
3.2.3 日付(Date)	O	O	O	O	O	O
3.3 メタデータスキーマ (Metadata Scheme)	O	M	M	M	M	M
3.4 言語(Language)	O	O	O	O	O	O
4.0 技術的事項(Technical)	O	M	M	M	M	M
4.1 フォーマット(Format)	O	M	M	M	M	M
4.2 サイズ(Size)	O	O	O	O	O	O
4.3 格納場所(Location)	O	M	M	M	M	M
4.4 必要条件(Requirement)	O	O	O	O	O	O
4.4.1 複数条件(OrComposite)	O	O	O	O	O	O
4.4.1.1 タイプ(Type)	O	O	O	O	O	O
4.4.1.2 名前(Name)	O	O	O	O	O	O
4.4.1.3 最低バージョン (MinimumVersion)	O	O	O	O	O	O
4.4.1.4 最高バージョン (MaximumVersion)	O	O	O	O	O	O
4.5 導入方法(InstallationRemarks)	O	O	O	O	O	O
4.6 その他のシステム要件(Other Platform Requirements)	O	O	O	O	O	O
4.7 継続時間(Duration)	O	O	O	O	O	O
5.0 教育的事項(Educational)	O	O	O	O	O	O
5.1 情報交換タイプ(Interactivity Type)	O	O	O	O	O	O
5.2 学習資源タイプ(Learning Resource Type)	O	O	O	O	O	O
5.3 対話性(Interactivity Level)	O	O	O	O	O	O
5.4 学習効率(Semantic Density)	O	O	O	O	O	O
5.5 利用者の種別(Intended End User Role)	O	O	O	O	O	O
5.6 利用環境(Context)	O	O	O	O	O	O
5.7 おもな年齢層(Typical Age Range)	O	O	O	O	O	O
5.8 難易度(Difficulty)	O	O	O	O	O	O
5.9 学習時間(Typical Learning Time)	O	O	O	O	O	O
5.10 記述(Description)	O	O	O	O	O	O
5.11 言語(Language)	O	O	O	O	O	O
6.0 権利(Rights)	O	M	M	M	M	M
6.1 費用(Cost)	O	M	M	M	M	M
6.2 著作権および制約(Copyrights and Other Restrictions)	O	M	M	M	M	M
6.3 記述(Description)	O	O	O	O	O	O

7.0 他のオブジェクトとの関連 (Relation)	○	○	○	○	○	○
7.1 種別(Kind)	○	○	○	○	○	○
7.2 資源(Resource)	○	○	○	○	○	○
7.2.1 識別子(Identifier)	○	○	○	○	○	○
7.2.1.1 カタログ(Catalog)	○	○	○	○	○	○
7.2.1.2 エントリ(Entry)	○	○	○	○	○	○
7.2.2 記述(Description)	○	○	○	○	○	○
8.0 注釈(Annotation)	○	○	○	○	○	○
8.1 情報(Entity)	○	○	○	○	○	○
8.2 日付(Date)	○	○	○	○	○	○
8.3 記述(Description)	○	○	○	○	○	○
9.0 分類体系(Classification)	○	M	M	M	M	○
9.1 目的(Purpose)	○	M	M	M	M	○
9.2 分類体系へのパス(Taxon Path)	○	○	○	○	○	○
9.2.1 分類体系の名前(Source)	○	○	○	○	○	○
9.2.2 各分類項目(Taxon)	○	○	○	○	○	○
9.2.2.1 識別子(Id)	○	○	○	○	○	○
9.2.2.2 エントリ(Entry)	○	○	○	○	○	○
9.3 記述(Description)	○	M	M	M	M	○
9.4 キーワード(Keyword)	○	M	M	M	M	○

表 3.1.4.6a: SCORM メタデータアプリケーションプロファイル要求条件

4 章

シーケンシングの概要

DRAFT

このページは空白である。

DRAFT

4.1. 概要

このドキュメントの多くの部分は、LMS が独立した学習アクティビティを一貫した順序に並べるといったように、作成された学習経験の意図する動作を表す方法を定義した IMS SS 仕様に基づいている。学習設計者やコンテンツ開発者は、実行時に配信する学習アクティビティを決定し、それに対応した学習資源（SCO や SCA）を起動するための、相対的な順序と条件を（コンテンツアグリゲーションにより）宣言する。このドキュメントは、準拠システムが実装しなくてはならない動作と機能を定義している。コンテンツアグリゲーションを通じて、学習アクティビティと学習者のやりとりの結果に基づく学習アクティビティの分岐やフローを記述するルールを取り入れている。この意図した教育戦略の表現は、手動で、あるいは、このプロファイルに準拠する出力を生成するオーサリングツールを用いて作成される。コンテンツ開発者はコンテンツシーケンスを生成し記述する方法を知る必要があるが、オーサリングシステムはこのプロファイルで提示されたモデルの詳細を隠蔽してもよい。シーケンシングの表現は、学習アクティビティを学習者に配信するように設計された LMS 間でやり取りされる。シーケンシング情報を解釈し、指定されたシーケンシング動作を実行する LMS のコンポーネントは「シーケンシングエンジン」と呼ばれる。

このドキュメントのベースとなる IMS SS 仕様は *simple* と呼ばれている。なぜならそれは広く使われる限られたシーケンシング動作を含むからで、規格そのものが単純だからではない。IMS SS は全てを含むわけではない。実際 IMS SS は、人工知能に基づくシーケンシング、スケジュールに基づくシーケンシング、外部のクローズドなシステムやサービスからのデータを要求するシーケンシング（例えば埋め込まれたシミュレーションのシーケンシング）、協調学習、カスタマイズされた学習、または、多重並列学習アクティビティの同期化について、必ずしも排除していないが、言及もしていない。

IMS SS は学習者の役割だけを認識し、インストラクタ、メンタ、同僚など他の登場人物を利用したり依存するシーケンシング機能を定義しない。このアプリケーションプロファイルは他の登場人物を含む文脈での利用を禁止しないが、他の登場人物の役割や他の登場人物の参加の結果となるシーケンシング動作は定義しない。加えて、コンテンツのルックアンドフィールや表示スタイルなどの課題も定義されない。

SCORM ver.1.3 アプリケーションプロファイルは、コンテンツアグリゲーションパッケージのための SCORM パッケージングアプリケーションプロファイルの拡張により、異なるランタイムコンポーネント間や LMS 間でのシーケンシング記述について交換するための外部表現を提供するものである。従って、シーケンシングは、コンテンツパッケージ規格と同様のコンテンツ構成とツリー構造に基づいている。

4.2. シーケンシング用語

以降の節では、このドキュメントで導入したモデルを理解するために不可欠なシーケンシングの語彙について詳述する。各記述はIMS SSの用語解釈に基づいている。

4.2.1. 学習アクティビティ / アクティビティ

IMS SS仕様は「学習アクティビティ」の概念に基づいている。学習アクティビティ（図4.2.1a）は、大雑把に言えば、意味のある教育単位である。学習アクティビティは学習資源を利用する場合もあるし、複数の子アクティビティからなる場合もある。

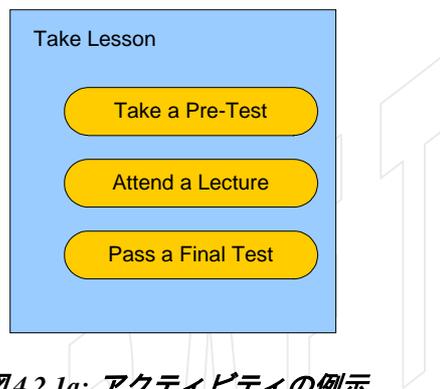


図4.2.1a: アクティビティの例示

上の図で、「Take Lesson（レッスンを取る）」アクティビティは、「Take a Pre-Test（事前テストを受ける）」、「Attend a Lecture（講義に出る）」、「Pass a Final Test（最終試験にうかる）」の三つの子アクティビティからなる。これは、定義されたアクティビティ階層が存在すること、および、「Attend a Lecture（講義に出る）」アクティビティを実行せずに「Take a Pre-Test（事前テストを受ける）」ことはできないことを暗示している。子アクティビティは、さらに子アクティビティから構成される場合もあるし、SCOやSCAなどの学習資源を参照する場合もある。

LMSは、学習者の前の学習アクティビティにおける進捗、学習者の意図、および、作成されたシーケンシング指示に基づいて実行時に決められる順序で学習アクティビティを配信する。学習アクティビティは以下の特性を有する。

- 学習アクティビティは明確な開始と終了を有する。
- 学習アクティビティは明確に定義された完了および習得条件を有する。
- 学習アクティビティは任意の深さの入れ子となった子アクティビティを含むことができる。
- 学習アクティビティは、親アクティビティが存在するならば、そのコンテキストの中で生じる。

-
- 学習アクティビティは、対応する学習資源（SCO ないし SCA）を有する場合も有しない場合もある。

学習者は、学習アクティビティのコンテキストの中で学習資源を経験する。学習者が学習アクティビティを試行する時、学習者は、その学習アクティビティに対応した学習資源が存在すれば、その学習資源を経験する。学習者が経験する一連の学習資源を学習経験と呼ぶ。学習資源の学習アクティビティへの対応付けは SCORM コンテンツアグリゲーションを通して行われる。

- コンテンツアグリゲーションは学習アクティビティの集合からなる。
- SCA と SCO は学習アクティビティに対応付けられる。

4.2.2. アクティビティのトラッキング

学習者が試行する各アクティビティについて、そのアクティビティは対応するトラッキング状態データを持たなくてはならない。学習者の学習資源とのやり取りは、その学習資源に対応した学習アクティビティのトラッキングデータに影響を及ぼすことができる。トラッキング状態データは種々のシーケンシングプロセスで用いられ、その動作に影響を及ぼす（詳細は 7.1.1 を見よ）。

4.2.2.1 データの保持

シンプルシーケンシングは、特定の学習者のアクティビティツリーに対する複数のシーケンシングセッションをまたいで、例えば、学習経験をまたいで、あるいは、複数のログインセッションをまたいで、データ（例えばトラッキングデータ）をどのように保持するかを指定しない。少なくともコンテンツアグリゲーションに対する現在の試行が終わるまで、制御、トラッキング、状態データを保持することが必要である。このような試行は、ひとつないし複数のシーケンシングセッションを含む場合がある。このような試行を超えてデータを保持するかどうかは LMS の方針に依存し、このような方針はアプリケーションプロファイルの範囲外である。

4.2.2.2 学習目標

IMS SS 仕様では学習目標は学習アクティビティと独立している。学習目標はローカルないしグローバルなスコープを有するデータ項目の集合を表し、各々は習得状態と習得度を有している。シンプルシーケンシングは学習目標の解釈方法（例えば、それがコンピテンシなのか、習得度なのか、単なる共有値なのか？）について仮定を置かない。アクティビティはひとつ以上の対応するローカル学習目標を有することができ、また、複数のグローバル共有学習目標を参照できる。複数のアクティビティは同じグローバル目標を参照できる、つまり、データ値を共有できる。アクティビティがグローバル共有目標を参照する方法には、二つの制限がある。

-
1. ひとつのローカル学習目標はひとつのグローバル共有目標だけから学習目標データを取得（‘read’）できる。
 2. あるアクティビティに対して定義されたローカル学習目標の集合において、複数のローカル学習目標が同じグローバル共有目標に学習目標データを設定（‘write’）することはできない。

ローカルおよびグローバル学習目標の ID の解決方法は指定されていない。グローバル共有目標は特定の学習者と学習目標 ID に固有である。グローバル共有目標は、単一のコンテンツアグリゲーション内、もしくは、複数のコンテンツアグリゲーション間で共有される場合がある。グローバル共有目標は、学習者が LMS 内に存在している限り、保持されていなくてはならない。アクティビティは複数の学習目標を参照し、これによって、アクティビティが副学習目標を持つ仕組みが提供される。しかし、アプリケーションプロファイルは、アクティビティに対応した複数学習目標のセマンティックスや意味について何も仮定しない。

4.2.3. 能動的 / 受動的コンテンツ

シンプルシーケンシングはトラッキング状態データモデルの値に基づいてシーケンシングの制御を行う。シンプルシーケンシングは、能動的コンテンツと受動的コンテンツを区別し、双方のコンテンツを個々のアクティビティごとに区別して取り扱う。能動的コンテンツは、SCORM ランタイム環境データモデル（詳細は 9 章参照）を通して、トラッキング状態データモデルの要素を設定しなくてはならない。受動的コンテンツに対しては、シンプルシーケンシングは、定義された既定値に基づき、自動的に特定の値をトラッキングモデルに設定する。

4.2.4. アクティビティツリー

SCORM コンテンツアグリゲーションモデルは、学習コンテンツの階層構成を提供する構造を定義している。これは、コンテンツパッケージ構成要素で表現されているコンテンツアグリゲーションの形式でなされている。

階層構造中の各アイテムは教育的に意味のある教育単位を表す。アイテムは任意の深さの入れ子となることができ、それぞれに適用された学習分類ラベルを有することができる。例えば、アイテムは、コース、モジュール、ユニット、レッスン、などを表すことができる。シーケンシングの観点からは、アイテムは独立した学習アクティビティに対応する。シーケンシング定義要素は、望ましい学習経験に合致した特定のシーケンシング実行時動作を定義するために、コンテンツ作成者によりアクティビティに適用される。「アクティビティ」はこの規格においては、学習経験のある部分を指す一般用語である。

この階層構造は、伝統的にコンテンツ構成の観点からコンテンツの交換の目的で指定されてきた。これが、コンテンツパッケージングマニフェストオーガニゼーション要素の概念の目的である。シーケンシング定義モデルは、実行時動作同様、意味のあるコンテンツ構造のコンテキストの中でのみ適切な結果をもたらすため、規格が共通の概念構成を定義することが必要である。

SCORM Version 1.3 アプリケーションプロファイルと IMS SS 仕様は、アクティビティの階層構成を「アクティビティツリー」と呼ぶ。アクティビティツリーはコンテンツアグリゲーションの（実装固有な）内部表現である。実装は、内部ではコンテンツアグリゲーションをツリーとして表現する必要は無い。その目的はシーケンシング動作記述の概念的なアンカーポイント（共通認識）を提供することである。アクティビティツリーは、作成された学習アクティビティの階層を表す一般的な概念構成と考えることができる。

アクティビティツリーは以下のように使用される。

1. アクティビティツリーは、コンテンツ設計、オーサリング、集約過程の結果得られるコンテンツ構造を表す。最終的に、アクティビティツリーは SCORM コンテンツパッケージのコンテンツアグリゲーションとして表わされる。オーサリングツールは、例えば、この階層を独自の形式で表現する内部データ構造を実装しているかもしれない。この構造は、作成者や設計者が意図した学習経験を作成するために取り入れた、どのような教育設計過程や方法からも得られる。
2. SCORM LMS はコンテンツアグリゲーションをアクティビティツリーに変換する。アクティビティツリーは、LMS が実装する、学習者ごとの階層中の各アクティビティのトラッキング状態情報を含む、定義された学習アクティビティの階層型の内部表現を反映したデータ構造を表わす。この文脈で、コンテンツ作成者が定義するシーケンシング定義情報は、LMS が学習アクティビティの相対順序を判断する方法を決定するとともに、学習者が条件に基づいて試行する学習アクティビティの選択的確性を決定する。

SCORM LMS は SCORM コンテンツパッケージを読み込み、そのコンテンツアグリゲーションを各学習者ごとの「アクティビティツリー」に変換する。シーケンシングを実装するシステムが、「アクティビティツリー」の独自の内部表現を有することが予期されるが、それは要求条件ではない。

アクティビティツリーにより、このアプリケーションプロファイルで、シーケンシングアルゴリズムや動作などの情報およびプロセッシング要件を、実装に依存しない方法で記述することが可能となる。繰り返しとなるが、オーサリングツールや LMS がアクティビティツリーを実装するよう義務付けたり、全ての教育設計手法が最終成果物としてアクティビティツリーに焦点を合わせるように修正されることは、SCORM Version 1.3 アプリケーションプロファイルの意図

するところではない。むしろ，アクティビティツリーは，階層的学習アクティビティとそれに対応するシーケンシング情報の具体例を表わす一般用語である。

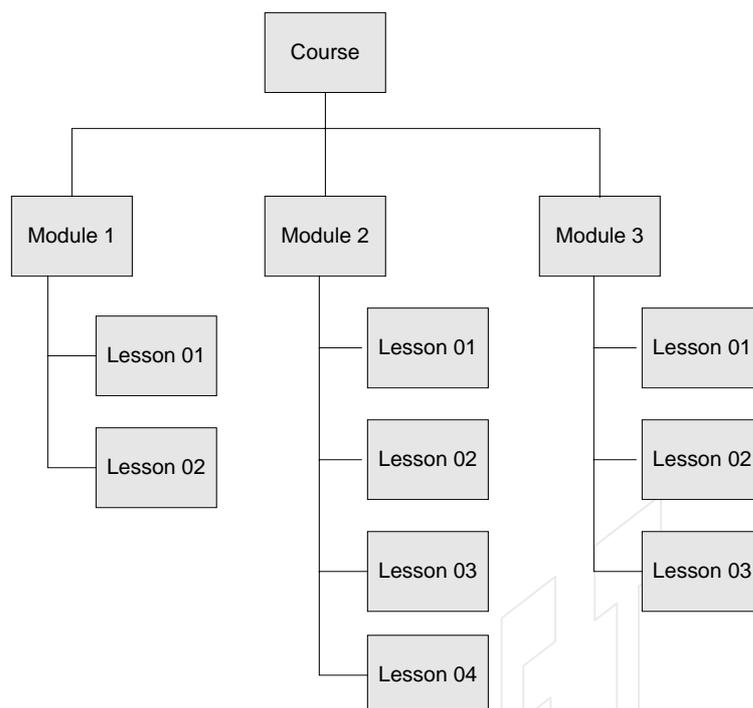


図4.2.4a: アクティビティツリーの表現

4.2.5. クラスタ

クラスタは子アクティビティからなるアクティビティである。クラスタはひとつの親アクティビティとその最初のレベルの子を含むが，子の子孫は含まない。クラスタはコンテンツアグリゲーションとシーケンシングの基本的な構成要素と考えることができる。クラスタは組み合わされてアクティビティツリーを構成する。クラスタの親アクティビティは，クラスタのシーケンシング戦略に関する情報を有する。クラスタの子で葉のアクティビティは，定義された戦略に従って起動される学習資源を有する。

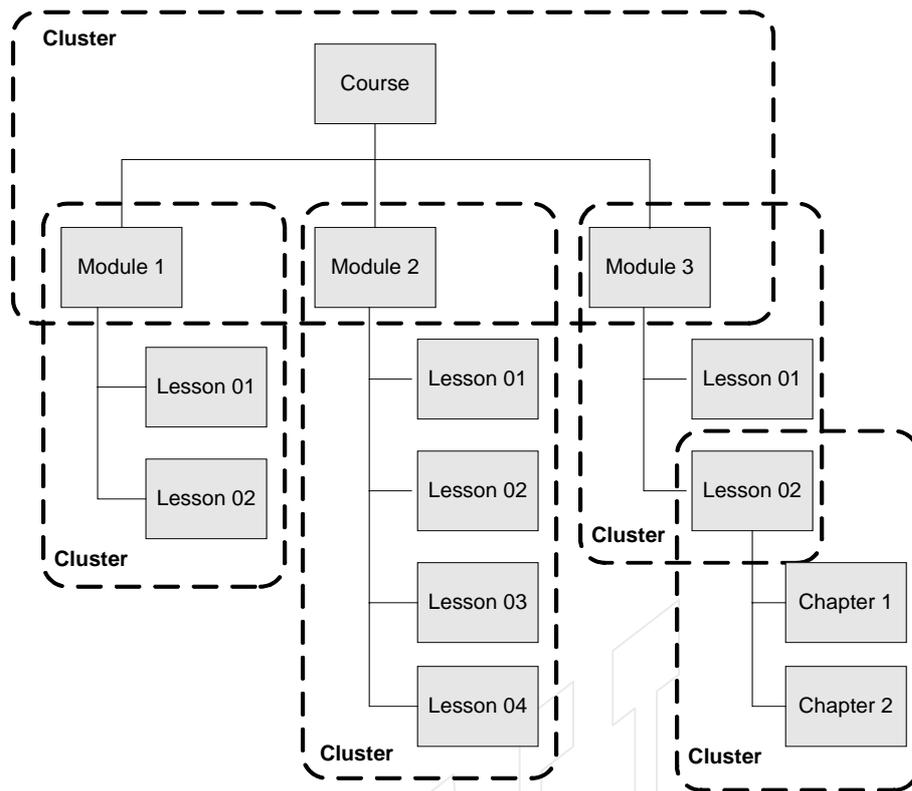


図 4.2.5a: クラスタの表現

図 4.2.5a は三つのモジュールを有する単純なコースを表わす。コースは、三つのモジュール全てを含むがレッスンは含まないクラスタの親である。各モジュールは同様に自身とその子からなるクラスタ自身の親である。モジュール 1 は、レッスン 01 と 02 からなるクラスタの親である。

4.2.6. アクティビティの経過のトラッキング

以下に示す学習経験の時間モデルは、時間経過に渡って学習経験を捕捉するものである。学習経験は、学習者に対して学習アクティビティへの試行と言う形で提示される。アクティビティの試行はひとつ以上のセッションに渡って遂行される。これらの用語は SCORM Version 1.3 アプリケーションプロファイルを通して使用され、以下に詳述されている。

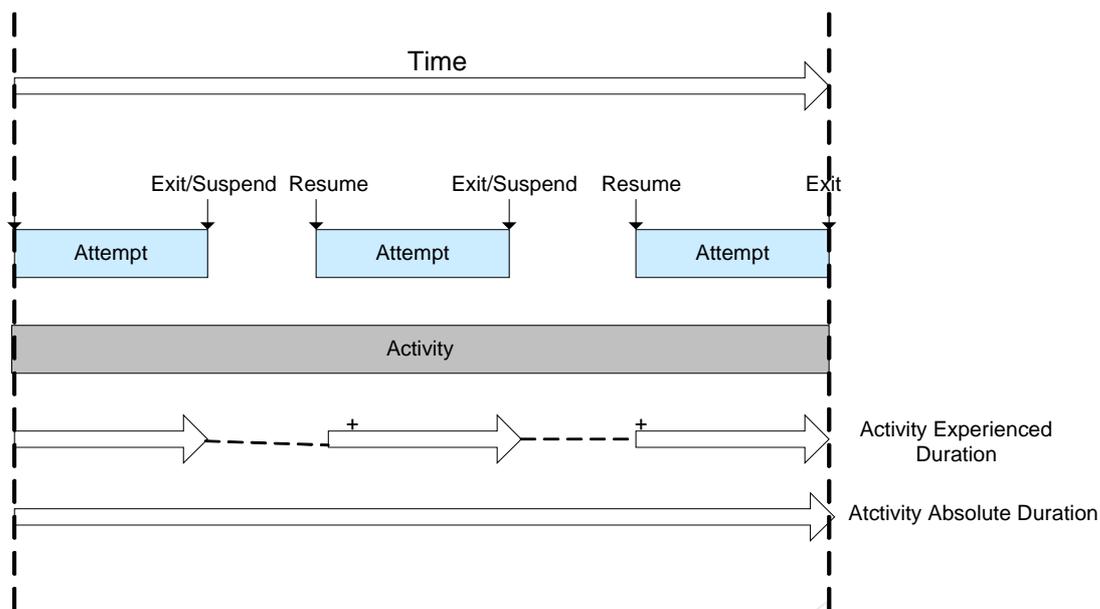


図4.2.6a: アクティビティ経過のトラッキング

4.2.6.1 試行

試行 (attempt) は学習アクティビティを完了するための試み、と定義される。アクティビティの試行は、常に、親アクティビティの試行のコンテキストの中で生じる。常にただひとつの葉アクティビティが試行され、その時、葉アクティビティの（ルートまでの）全ての祖先も試行される、ということに注意しなくてはならない。葉アクティビティが試行されているとき、アクティビティに対応する学習資源が起動していると仮定することができる。一回の試行でアクティビティを完了することは、常に可能ではない。学習者が学習アクティビティを中断し、後で再開しなくてはならない状況が多々ある。たいていの場合、これは現在の試行の継続で、新しい試行ではない。学習者によるアクティビティの試行の結果、あるいは、外部の管理動作の結果、アクティビティのトラッキング状態が変更されうる（トラッキング状態情報については7.1.1参照）。いずれかのアクティビティのトラッキング状態が変化するとき、この変化の親アクティビティへの影響は、アクティビティの起動の評価を含む後続するいかなるシーケンシングプロセスよりも前に利用可能とならなくてはならない。アクティビティの試行がいかなる理由にせよ終了したとき、アクティビティツリーの残りの部分に対するトラッキング状態の変化の影響は、次のナビゲーション要求が処理される前に利用可能とならなくてはならない。

4.2.6.2 アクティビティの中断と再開

アクティビティの試行は、中断し、後に再開することができる。中断されたアクティビティの再開は、そのアクティビティに対する新しい試行とはみなされない。中断中に他のアクティビティを試行することができる。ひとつ以上のアクティビティが同時に中断される場合もある。

現在のシーケンシングセッションは、中断された後、学習者が経験した最後のアクティビティから再開される。

4.2.6.3 シーケンシングセッションの開始と停止

シーケンシングセッションは、オーバオールシーケンシングプロセスが始まってから終わるまでの時間である。この期間は、コンテンツアグリゲーションのルートアクティビティの試行の絶対試行経過時間に等しい。

シンプルシーケンシングは、どのナビゲーション要求がシーケンシングセッションを開始できるかだけを指定していて、それらのナビゲーション要求がいつどのように誘起されるかは指定していない。一般に、LMS は特定のシステムイベント、例えばコースへのログイン、を認識して Start ナビゲーション要求を発行する。もし、先立つシーケンシングセッションが Suspend All ナビゲーション要求で終了していたら、LMS は Start の代わりに Resume All ナビゲーション要求を使わなくてはならない。場合によっては Start ナビゲーション要求が成功せず、LMS は Choice ナビゲーション要求を呼び出す仕組みを提供する必要がある。

シーケンシングセッションは、Exit シーケンシング要求がアクティビティツリーのルートアクティビティで実行されたとき、終了する。一般にこれが生じるのは、Exit All ナビゲーション要求（すなわち logout）か、終了動作シーケンシングルールがアクティビティツリーのルートに適用されたときである。

4.2.7. シーケンシング定義モデル

シーケンシング定義モデルは、コンテンツ開発者がシーケンシング動作に影響を与えるのに利用できる要素の集合を定義している。準拠 LMS は、これら全ての定義の結果得られる動作をサポートしなくてはならない。定義モデルにより、特定の要素を学習アクティビティに対応付けることができる。各情報モデル要素は、陽にコンテンツ開発者が指定する値が無い場合に仮定される既定値を有する。

4.2.8. コンテンツタイプ

SCORM シーケンシングアプリケーションプロファイルは学習コンテンツと学習目標のタイプにある意味で非依存であり、全てのタイプのコンテンツをシーケンシングするために利用できる。例えば、コンテンツは、単純な静的 Web ページ、あらゆる種類（例えば、DOC、PDF ファイル）の Multipurpose Internet Mail Extensions (MIME) 資源、サービスや動的に生成されるオブジェクトへのサービスやプロキシ、などを包含する。実際、コンテンツは SCO で使用される SCORM 通信 API を使用しなくてもよい。いくつかのトラッキングモデル値は SCORM ランタイムデータモデル要素と一致するが、これはシーケンシングの要

求条件ではない。しかし，SCORM Version 1.3 アプリケーションプロファイルで可能となるもっと複雑なモデルを実行するには SCORM ランタイムを使用しなくてはならない。

DRAFT

このページは空白である。

DRAFT

5 章

シーケンシング定義モデル

DRAFT

このページは空白である。

DRAFT

5.1. SCORM シーケンシング定義モデル

SCORM シーケンシング定義モデルは、IMS SS 定義モデルに直接基づいている。SCORM シーケンシング定義モデルは、IMS SS 仕様にアプリケーションプロファイル特有の追加の動作と制限を規定したものである。

この定義モデルは、コンテンツ制作者が特定のシーケンシング動作を規定するために使用することのできる要素の集合を定める。準拠 LMS はこれらすべての要素の定義に応じた動作をサポートしなければならない。定義モデルを使って、個々の要素を学習アクティビティに関連付けできる。個々の情報モデルの要素には既定値が設定され、コンテンツの作者により明示的にすべてを提供しなくても大丈夫なようにしてある。

シーケンシング定義モデルは以下のカテゴリーに分けて定義されている：

- シーケンシングコントロールモード
- シーケンシングルール
- 制限条件
- 補助リソース
- ロールアップルール
- 学習目標
- 学習目標マップ
- ロールアップコントロール
- 選択コントロール
- ランダム化コントロール
- 配信コントロール

5.1.1. シーケンシングコントロールモード

シーケンシングコントロールモードを使えば、必要に応じて、コンテンツの作者が望む学習経験を反映したシーケンシング動作をコントロールできる。コントロールモードは以下のケースで使用される：

- ナビゲーションリクエストの処理（ナビゲーションリクエスト動作 - Navigation Request Behavior）中に、リクエストを有効なシーケンシングリクエストに変換するかどうかを決定する。
- さまざまなシーケンシングのサブプロセス中に、これらのサブプロセスが返却する結果的な配信リクエストに反映する。そして
- さまざまなシーケンシングのサブプロセス中に、トラッキング状態データモデル要素の状態をどのように取り出すかに影響する（*Use Current Attempt Objective Information* と *Use Current Attempt Progress Information*）。

このコントロールモードを定義することで、コンテンツの開発者は、クラスタに対するシーケンシング動作を制御できる。ひとつのクラスタに対して、同時に複数のモードを適用することもでき、これにより複数のシーケンシングコントロール動作の組み合わせが可能である。

5.1.1.1 Sequencing Control Choice (シーケンシングコントロール Choice)

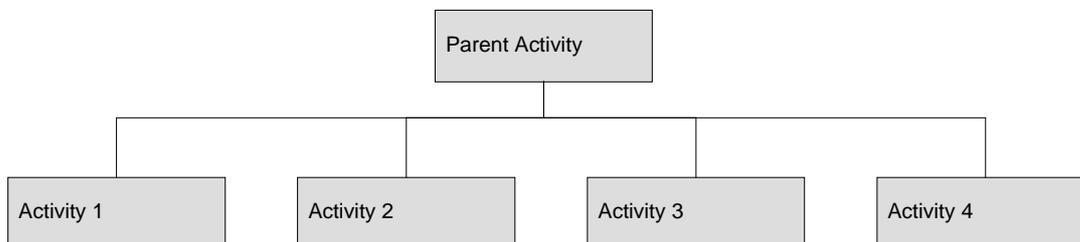


図5.1.1.1a: Sequencing Control Choice

Sequencing Control Choice コントロールモードでは、学習者はどのアクティビティをどのような順序でも選択できる。親の *Sequencing Control Choice* が True なら、どの子アクティビティでも (Any) Choice ナビゲーションリクエストに対する正当な対象となる。図 5.1.1.1a のケースでは、親のアクティビティのシーケンシングコントロールが True に設定されていれば、子アクティビティ 1-4 は Choice ナビゲーションリクエストに対する正当な対象となる。親のアクティビティは、さらにその親のアクティビティの Choice シーケンシングコントロールが True に設定されていない限り、正当な対象にはならない。

他のシーケンシングルールによる配信禁止が指定されていない限り、学習者があるアクティビティを選択すると、このアクティビティに関連する学習リソースが起動されなければならない。Choice コントロールモードが有効な場合、学習者が Choice ナビゲーションイベントを発生するための手段が提供されていると仮定される。Choice ナビゲーションイベントにより、指定された目標アクティビティに対する Choice シーケンシングリクエストが発生する。このことからわかるように、Choice ナビゲーションイベントを可能とするために、LMS は「メニュー」、「マップ」、「目次」などのユーザインターフェースを用意しておく必要がある。

Sequencing Control Choice コントロールモードは、明示的に無効 (Choice が False) と指定されない限り、常に有効状態である。

上述のとおり、その親が True と定義されている *Sequencing Control Choice* を持つすべてのアクティビティは Choice ナビゲーションリクエストに対する正当な対象となる。学習者が下位のアクティビティを持つ、すなわち親のアクティビティ (コンテンツアグリゲーションモデルで定義したクラスタの場合でかつ学習リソースを持たない) を選択すると、シーケンシングプロセスがとり得る動作は以下の 2 つのいずれかになる：

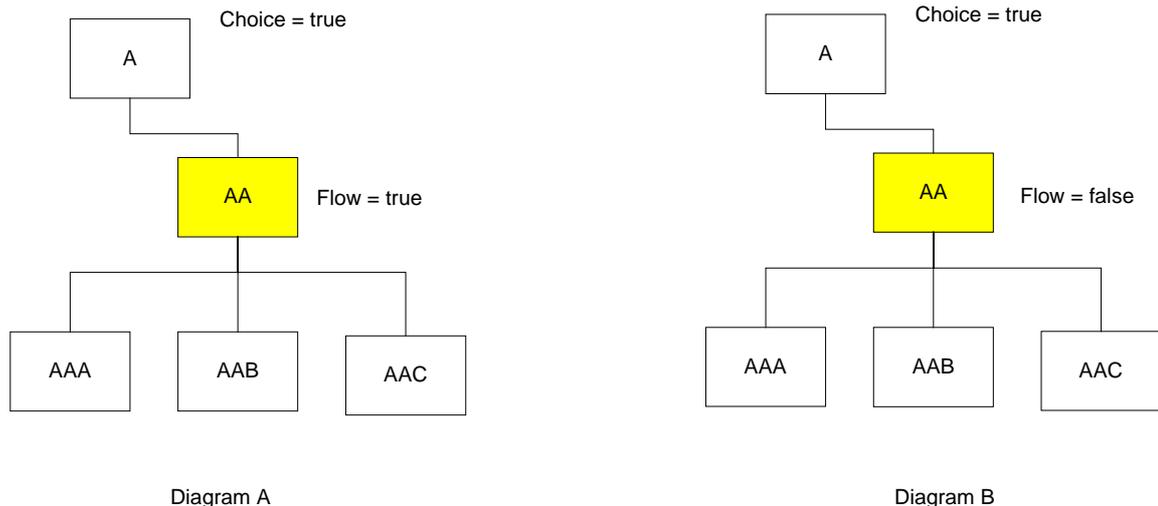


図5.1.1.1b: 「選択」の説明

上図中の図 A では、Choice ナビゲーションリクエストの対象（アクティビティ AA）は、その *Sequencing Control Flow* が True に定義されている。これは、学習リソースをもつ子アクティビティが見つかるまで、あらかじめ定義された順序で子アクティビティの集合をトラバースする必要がある、ということの意味し、その（見つかった）子アクティビティは配信のために指定される。

逆に上図中の図 B では、Choice ナビゲーションリクエストの対象が、*Sequencing Control Flow* が False として定義されている。これは、シーケンシングプロセスは *interstitial*（わずかの隙間の意）プロセス（将来説明される予定）に従う必要がある、ということの意味する。

アクティビティが終了しても、学習者が他のアクティビティを選択するか、またはメインのアクティビティを終了(quit)するまでは何も起こらない。アクティビティが実行されている間に、学習者が他のアクティビティを選択した場合、（詳細は「シーケンシングコントロール Choice Exit」にあるが、）まず現在のアクティビティの学習リソースが終了され（terminated）、そのあと選択されたアクティビティが実行される。

5.1.1.2 Sequencing Control Choice Exit（シーケンシングコントロール Choice Exit）

Sequencing Control Choice Exit、以下単に ChoiceExit と呼ぶ、コントロールモードは、LMS のシーケンシングエンジンの実装に対し、Choice ナビゲーションリクエストが当該のアクティビティの子孫ではないアクティビティを対象にでき、結果として当該のアクティビティを終了できることを指示するのに使用しなければならない。ChoiceExit はアクティブなアクティビティにだけ適用できる。アクティビティに特別の定義がなされていない限り、この既定値は True である。このことはアクティビティがアクティブな間は、学習者には子孫以外のアクテ

ィビティを対象にする Choice ナビゲーションリクエストを発信できる環境が設定されることを示す。

5.1.1.3 Sequencing Control Flow (シーケンシングコントロール Flow)

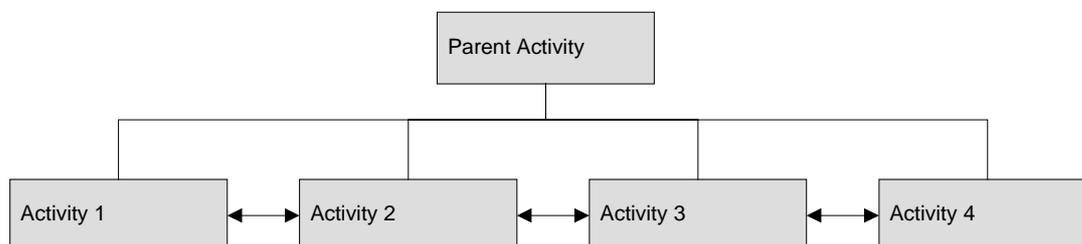


図5.1.1.3a: Sequencing Control Flow

Sequencing Control Flow コントロールモードは、あるアクティビティに対して定義されている子のアクティビティを通してガイド付のシーケンシングを可能とするものである。図 5.1.1.3a のケースでは、親のアクティビティの *Sequencing Control Flow* が True に設定されていれば、アクティビティ 1-4 は (「次へ」と「前に」の) ガイド付のパスによるシーケンシングが実装される。

Sequencing Control Flow コントロールモードに対して、LMS は *Continue* (次へ) と *Previous* (前へ) のシーケンシングコントロールが可能となるような処理を行う必要がある。もし *Sequencing Control Flow* コントロールモードが一つのアクティビティクラスに対して指定され、コンテンツの作者が、コンテンツ内にナビゲーションイベントを発生させる独自のメカニズムを提供するように設定していない場合、LMS は学習者が *Continue* で次のアクティビティに移動し、*Previous* で前のアクティビティに移動できるメカニズムを提供する必要がある。

5.1.1.4 Sequencing Control Forward Only (シーケンシングコントロール Forward Only)

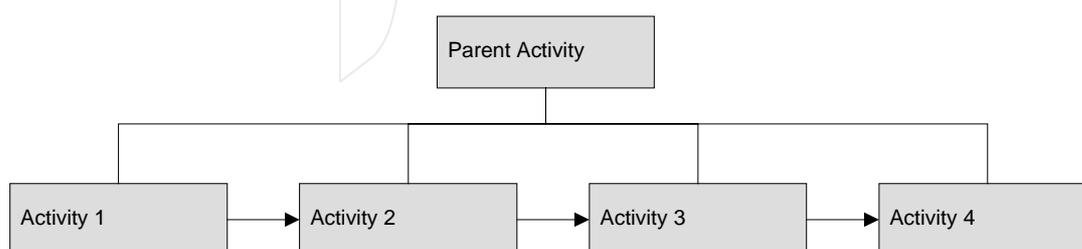


図5.1.1.4a: Sequencing Control Forward Only

Sequencing Control Forward Only、以下単に *ForwardOnly* と呼ぶ、要素は、それが定義されたノードの子の間のシーケンシングを制限することを示している。この要素は真偽値を持つ。アクティビティに特に定義されていなければ *ForwardOnly* の既定値は False である。あるノードに対する *Sequencing Control Forward Only* が True なら、ノードに属する子に対するトラバースは常に前方に行われる。たとえばツリーに対するトラバースの順で見て、次のノードから

Previous シーケンシングリクエストの結果として再度同じノード（クラスタ）が選択される場合，（クラスタの）最後の子からではなく，最初の子ノードが最初にトラバースされる．図 5.1.1.4a では，親のアクティビティの Sequencing Control Forward Only は True に設定されている場合，子のアクティビティ（アクティビティ 1-4）は（次へ，次への）連続モードでのみ経験できるようになる．この場合，学習者に Previous 方向へ移動してのアクティビティの学習経験は許されない．

親のノード（アクティビティ）にルールが定義されている場合，以下の要求に従う必要がある：

Sequencing Control Forward Only が True の場合：

1. *Sequencing Control Flow* が True に設定されていると，クラスタ内の子間に対する逆方向のナビゲーションは許されない．
2. *Sequencing Control Choice* が True に設定されている場合，目標のノードが現在配信されている子のひとつ前の子ノードの場合，シーケンシングリクエストは無視される．

Sequencing Control Forward Only が False 値の場合（既定値）：

1. *Previous* と *Choice* のシーケンシングコントロールリクエストには何も制限が課せられない．

5.1.1.5 Use Current Attempt Objective Information（現試行学習目標情報の使用）

この要素は，アクティビティに対する現在の試行中に学習目標達成度情報が記録された場合のみ，アクティビティの子の学習目標達成度情報をルールの評価とロールアップの評価に使用することを（True または False 値により）指示する．既定値は True である．

Use Current Attempt Objective Information を False に設定すると，たとえ直前にすでに記録されていたとしても，子のアクティビティに対する最新の試行での学習目標達成情報が（再度）使用される，ことを指示することになる．

5.1.1.6 Use Current Attempt Progress Information（現試行進捗情報の使用）

この要素は，アクティビティに対する現在の試行中に試行進捗情報が記録された場合のみ，アクティビティの子の試行進捗情報をルールの評価とロールアップの評価に使用することを（True または False 値により）指示する．既定値は True である．

Use Current Attempt Progress Information を False に設定すると，たとえ直前にすでに記録されていたとしても，子のアクティビティに対する最新の試行での試行進捗情報が（再度）使用される，ことを指示することになる．

5.1.2. シーケンシングルール

IMS SS の仕様ではルールベースのシーケンシングモデルを採用している。ひとつのアクティビティに、0 個以上のルールを定義でき、実行時にこのルールはシーケンシングと配信の適切な時点で評価される。定義されたルールの評価をどのように実現するかは自由であるが、少なくとも、さまざまなシーケンシングプロセス中のある時点で定義されたシーケンシングルールが評価され、定義された動作が実現されるようにしておかなければならない（「第 7 章シーケンシング動作」参照）。あるアクティビティに対するシーケンシングルールの適用により、既定値のシーケンシングパスが変更されたり、アクティビティと子のアクティビティの両方またはいずれかの利用に影響を及ぼす。

シーケンシングルールは、一連の条件と、これが満足されたときにとられるアクションないし動作(*if [condition_set] then [action/behavior]*)から構成される。シーケンシングルール条件は、あるアクティビティの、*Objective Progress Information*、*Activity/Attempt Progress Information* などのトラッキング状態データモデル (Tracking Status Data Model) に基づくか、あるいは定義された *Limit Conditions* に関するアクティビティの状態に基づく。IMS SS の仕様ではトラッキングモデル (Tracking Model) を定義して、シーケンシングルール条件に従う要素を規定している。

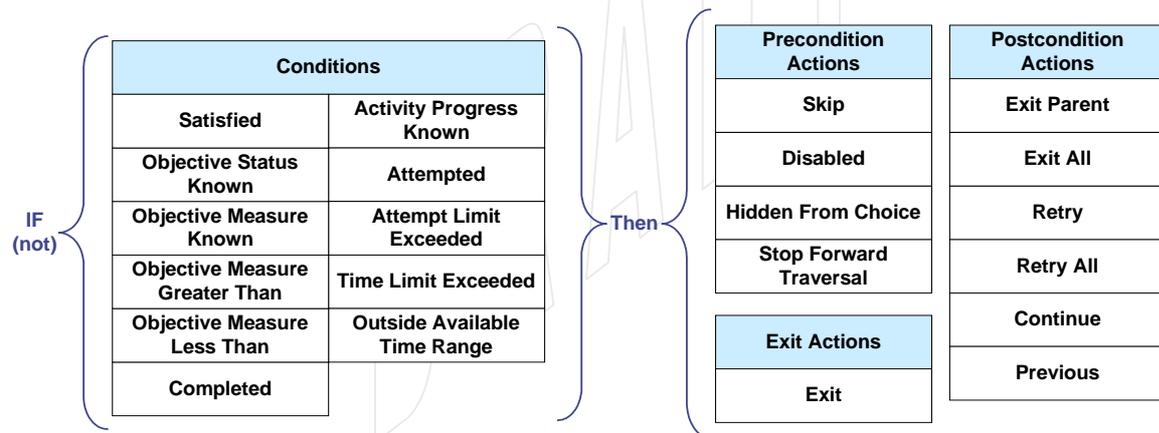


図 5.1.2a: 条件およびアクション/動作

5.1.2.1 シーケンシングルールの組み合わせ

個々の条件を組み合わせ、あるひとつの条件が True、または、全ての条件が True のとき、結果の動作が有効になるような条件の集合を作ることができる。デフォルトでは、シーケンシングルールの集合に対して修飾子が陽に定義されていないとき、シーケンシングルール集合中の全てのルールが評価される。

IMS SS 仕様では 2 種の修飾子 (*Condition Combination*) を定義している。

- All. ルール条件集合は、全てのルールアップ条件が True のとき、True と評価される。論理積 (logical and) として働く。
- Any (default value 訳注 All が既定値が正しい)。ルール条件集合は、いずれかのルールアップ条件が True のとき、True と評価される。論理和 (logical or) として働く。

以下の表は Condition Combination 情報モデル要素の仕様を示している。

No.	名称	説明	値空間	既定値
1	Condition Combination	<p>ルール条件を組み合わせるルールを評価する方法。</p> <ul style="list-style-type: none"> • All – すべてのルール条件が True のときのみルール条件が True と評価される (論理積 logical and)。 • Any – いずれかのルール条件が True のときにルール条件が True と評価される (論理和 logical or)。 	語彙	All

表5.1.2.1a: Condition Combination の説明

5.1.2.2 条件

条件要素には、シーケンシングルールが適用されるアクティビティのコンテキスト中で評価される単一の条件が含まれる。個々の条件要素を他の条件要素と組み合わせる一連の条件セットを作成できる。コンテンツの開発者は、この組み合わせられた条件要素のセットを使用して LMS に一連の条件の評価方法を指示することができる。条件要素は、IMS SS でのトラッキング状態モデル (Tracking Status Model) をベースにした、制限された語彙項目の集合から構成される：

条件	説明
Satisfied	アクティビティに関連付けられた学習目標 (Rule Condition Referenced Objective で示される) に対する Objective Progress Status が True で、かつ Objective Satisfied Status 値が True のとき、この条件は True と判定される。
Objective Status Known	アクティビティに関連付けられた学習目標 (Rule Condition Referenced Objective で示される) に対する Objective Progress Status が True のとき、この条件は True と判定される。
Objective Measure Known	アクティビティに関連付けられた学習目標 (Rule Condition Referenced Objective で示される) に対する Objective Progress Status が True で、かつ Objective Measure Status (Objective Progress value) が True のとき、この条件は True と判定される。
Objective Measure Greater Than	アクティビティに関連付けられた学習目標 (Rule Condition Referenced Objective で示される) に対する Objective Measure Status が True で、かつ Objective Normalized Measure が Rule Condition Measure Threshold より大きいとき、この条件は True と判定される。
Objective Measure Less Than	アクティビティに関連付けられた学習目標 (Rule Condition Referenced Objective で示される) に対する Objective Measure Status

	が True で，かつ <i>Objective Normalized Measure</i> が <i>Rule Condition Measure Threshold</i> より小さいとき，この条件は True と判定される．
Completed	アクティビティに対する <i>Attempt Progress Status</i> が True で，かつ <i>Attempt Completion Status</i> が True のとき，この条件は True となる．
Activity Progress Known	アクティビティに対する <i>Activity Progress Status</i> が True で，かつ <i>Attempt Progress Status</i> が True のとき，この条件は True となる．
Attempted	アクティビティに対する <i>Activity Progress Status</i> が True で，かつ <i>Activity Attempt Count</i> が正值(つまり，このアクティビティは試行されたことがある状態)のとき，この条件は True となる．
Attempt Limit Exceeded	アクティビティに対する <i>Activity Progress Status</i> が True で，かつ <i>Limit Condition Attempt Limit Control</i> が True で，さらに <i>Activity Attempt Count</i> がアクティビティに対する <i>Limit Condition Attempt Limit</i> の上限条件値に等しいか大きいとき，この条件は True となる．
Time Limit Exceeded	この条件は以下のすべての条件を満たすとき True となる： <ol style="list-style-type: none"> 1. アクティビティに対する <i>Attempt Progress Status</i> が True 2. アクティビティに対する以下の期間値： <ol style="list-style-type: none"> a. <i>Activity Absolute Duration</i> b. <i>Activity Experienced Duration</i> c. <i>Attempt Absolute Duration</i> d. <i>Attempt Experienced Duration</i> がのいずれかがアクティビティに対する以下の該当の期間制限条件値を超えるととき： <ol style="list-style-type: none"> e. <i>Activity Absolute Duration Limit</i> f. <i>Activity Experienced Duration Limit</i> g. <i>Attempt Absolute Duration Limit</i> h. <i>Attempt Experienced Duration Limit</i> 3. アクティビティに対する以下の制限条件コントロール値が True のとき： <ol style="list-style-type: none"> i. <i>Limit Condition Activity Absolute Duration Control</i> j. <i>Limit Condition Activity Experienced Duration Control</i> k. <i>Limit Condition Attempt Absolute Duration Control</i> l. <i>Limit Condition Attempt Experienced Duration Control</i>
Outside Available Time Range	この条件は以下のとき True となる： <ol style="list-style-type: none"> 1. 現在時刻が，アクティビティに対する該当の時間制限条件値 (<i>Limit Condition Begin Time Limit</i> , <i>Limit Condition End Time Limit</i>) 時刻より以前 (<i>Begin Time Limit</i>) または以後 (<i>End Time Limit</i>) のとき 2. アクティビティに対する，関連する上限条件コントロール値 (<i>Limit Condition Begin Time Control</i>, <i>Limit Condition End Time Control</i>) が True に設定される．

表5.1.2.2a: ルール条件の説明

5.1.2.3 Rule Condition Referenced Objective (ルール条件参照学習目標)

Rule Condition Referenced Objective は，該当のアクティビティに対して定義された一連の学習目標から，学習目標情報を必要とするルール条件の評価に使用する

る学習目標を決定するのに使用される。もしアクティビティに対するルールが明示的に識別子により学習目標を参照していないなら、既定値として、ルールはアクティビティに対するロールアップに關与する学習目標を参照する。これらのルール条件には以下のものがある:

- Satisfied
- Objective Status Known
- Objective Measure Known
- Objective Measure Greater Than
- Objective Measure Less Than

No.	名称	説明	値空間	既定値
2.2	<i>Rule Condition Referenced Objective</i>	アクティビティに關する条件評価時に使用される学習目標の識別子。	固有の識別子	なし

表5.1.2.3a: Rule Condition Referenced Objective

5.1.2.4 Rule Condition Measure Threshold (ルール条件習得度閾値)

ルール条件を定義することによりアクティビティに關して習得度閾値を指示できる。この閾値は以下のルール条件の評価時に使用される:

- Objective Measure Greater Than: [学習目標習得度 (objective measure)] > [習得度閾値 (measure threshold)]
- Objective Measure Less Than: [学習目標習得度 (objective measure)] < [習得度閾値 (measure threshold)]

この要素は、-1.0 から 1.0 の間の実数 (-1.0 と 1.0 を含む) で定義する。この閾値は「大なり(>)」と「小なり(<)」で (設定された値と) 比較演算が行われることに、コンテンツの開発者は注意しなくてはならない。「大なりまたは等しい(>=)」と「小なりまたは等しい(<=)」を定義するルール条件はない。

No.	名称	説明	値空間	既定値
2.3	<i>Rule Condition Measure Threshold</i>	習得度に基づく条件評価時に使用される値。	実数 [-1.0..1.0] 精度は少なくとも 4 桁の十進数。	0

表5.1.2.4a: Rule Condition Measure Threshold

5.1.2.5 Rule Condition Operator (ルール条件演算子)

Rule Condition Operator 要素は単項の論理演算子で個々の条件に適用できるものである。IMS SS の現在の仕様では、この要素に 2 種類の値を定義している:

- NO-OP: ルール条件がそのまま使用される (既定値)
- Not: ルールの評価時にルール条件の逆が適用される

No.	名称	説明	値空間	既定値
2.4	<i>Rule Condition Operator</i>	条件に適用される単項論理演算子 <ul style="list-style-type: none"> • <i>Not</i> – 該当条件の逆がルール評価に使用される。 • <i>NO-OP</i> – 対応する条件がそのままルール評価に使用される。 	語彙	NO-OP

表5.1.2.5a: Rule Condition Operator

5.1.2.6 Actions (動作)

Action 要素は、シーケンシングプロセス中に LMS に行わせたいアクション (Action) ないしは動作 (Behavior) を指定するものである。アクションは3種類の固有の評価順にカテゴリ分けされる：

- 事前条件動作 (Precondition Actions) : 配信するアクティビティを決定するためにアクティビティツリーのトラバースする際に適用される。
- 事後条件動作 (Postcondition Actions) : アクティビティの試行終了時に適用される。
- 終了動作 (Exit Actions) : 子孫のアクティビティの試行が終了した (terminates) 後に適用される。

No.	名称	説明	値空間	既定値
事前条件動作				
3	<i>Rule Action</i>	このルールが True に評価されたときにとられる望ましいシーケンシング動作： <ul style="list-style-type: none"> • <i>Skip</i> – アクティビティは <i>Flow</i> シーケンシングリクエスト中に配信される候補とはみなされない • <i>Disabled</i> – アクティビティはすべてのシーケンシングリクエストまたは配信リクエストの目標にはされない。 • <i>Hidden from Choice</i> – アクティビティは <i>Choice</i> のシーケンシングリクエストの目標とはみなされない。 • <i>Stop Forward Traversal</i> – シーケンシングリクエスト中のアクティビティからの「次に」方向のアクティビティツリー移動が停止される I 	語彙	Ignore
事後条件動作				
3	<i>Rule Action</i>	このルールが True に評価されるときに	語彙	Ignore

		とられる望ましいシーケンシング動作： <ul style="list-style-type: none"> • <i>Exit Parent – Exit Parent</i> 終了リクエストが行われる。 • <i>Exit All – Exit All</i> 終了リクエストが行われ、<i>Exit</i> シーケンシングリクエストを返す。 • <i>Retry – Retry</i> シーケンシングリクエストを返す。 • <i>Retry All – Exit All</i> 終了リクエストが行われ、<i>Start</i> シーケンシングリクエストを返す。 • <i>Continue – Continue</i> シーケンシングリクエストを返す。 • <i>Previous – Previous</i> シーケンシングリクエストを返す。 		
終了動作				
3	<i>Rule Action</i>	このルールが True に評価されるときにとられるシーケンシング動作： <ul style="list-style-type: none"> • <i>Exit</i> – 無条件でアクティビティが停止される。1 	語彙	Ignore

表5.1.2.6a: *Actions* の記述

5.1.3. 制限条件

コンテンツの開発者は、制限条件を定義することでアクティビティの配信を許可するかどうかに関して、一定の条件をコントロールできる。制限条件はアクティビティに関連付けされた Tracking Model に基づく条件である。定義された制限条件に一致するか、これを上回ると、該当のアクティビティは配信されなくなる。

制限条件は常にシーケンシングルールより優先される。たとえばシーケンシングルールがアクティビティの再試行を指定していて、最高試行回数の上限も設定されている場合、最高試行回数を超過するとシーケンシングルールよりも制限条件が優先されるため、アクティビティは再試行されなくなる。

同様に、制限条件は常にシーケンシングリクエストより優先される。たとえば Choice シーケンシングリクエストが利用時間帯制限を持つアクティビティに対するものであった場合、該当時間帯以外であれば、アクティビティは選択されても配信可能にならない。

次の章は一連の制限条件を定義したものである。制限条件は、コントロール変数と上限値の一对の値で定義される。コントロール変数は、制限条件がコンテンツの作者によって定義されているかどうかを指示するのに使用される。制限

値は文字通り上限値や範囲を指定するのに使用される。特定のいくつかのシーケンシング処理が（Controlとして）定義されていて、これにより制限条件が定義されているかどうかを判定する。

5.1.3.1 試行

コンテンツの作者が、ある学習アクティビティに対して学習者に許容する試行回数を制限したいケースがある。シーケンシングルールにより特定の学習アクティビティに最大試行回数を指定することができる。制限条件を設定したい学習アクティビティに関する *Limit Condition Attempt Limit* 要素に、正の整数値を指定することで、最大試行回数を指定する。コンテンツの作者が、この *Limit Condition Attempt Limit* 値を定義しない場合、該当のアクティビティに関する試行回数の制限はないことになる。

No.	名称	説明	値空間	既定値
1	<i>Limit Condition Attempt Control</i>	アクティビティに関する試行回数の制限が設定されていることを示す。 False 値の場合、アクティビティに関する試行制限はない。	論理型	False
2	<i>Limit Condition Attempt Limit</i>	アクティビティへの最大試行回数。ゼロ値の場合、アクティビティはアクセスできないことを示す。 <i>Limit Condition Attempt Control</i> 値が True 以外の場合、この値は不定値である。	正の整数	0

表 5.1.3.1a: *Limit Condition (Attempt) Excerpt from IMS SS*

IMS SS の制限条件の表(表 4.1.3.1a),に従うと、*Limit Condition Attempt Limit* が定義されていると、*Limit Condition Attempt Control* が True に設定される。逆に *Limit Condition Attempt Limit* が定義されていない場合、*Limit Condition Attempt Control* は False に設定される。

5.1.3.2 Attempt Absolute Duration (試行絶対経過時間)

コンテンツの作者が、ある学習アクティビティの一回の試行ごとの試行時間に制限を加えたいケースがある。*Attempt Absolute Duration Limit* 要素に最大試行時間値を指定することで、学習者がこのアクティビティに費やす最大時間を制限できる。この制限時間は、システムや学習者の動作時間にかかわらず、LMS がアクティビティに対する試行を始めるときから終了するまでの時間である。コンテンツの作者が学習アクティビティに対して、この *Attempt Absolute Duration Limit* を定義しない場合、該当のアクティビティには時間制限が設けられないことになる。

No.	名称	説明	値空間	既定値
3	<i>Limit Condition</i>	アクティビティに関する 1 回の試行時間の最	論理型	False

	<i>Attempt Absolute Duration Control</i>	大値が指定されていることを示す。 値が False の場合、アクティビティの試行に関する制限時間は設けられないことを示す。		
4	<i>Limit Condition Attempt Absolute Duration Limit</i>	アクティビティへの 1 回あたりの試行時間の最大値。この時間は、アクティビティが <i>active</i> な時間に適用される。アクティビティのアクティブとは、それがはじまってから終了するまでの時間で、これには <i>suspended</i> 状態であった時間も含まれる。ゼロ値は、該当のアクティビティがアクセスできないことを示す。 <i>Limit Condition Attempt Absolute Duration Control</i> が True 値以外の場合、この要素の値は不定値である。	時間 - 単位 (精度) は 0.1 秒	0.0

表5.1.3.2a: *Attempt Absolute Duration Limit*

IMS SS の上限条件の表(表 4.1.3.2a),に従うと, *Attempt Absolute Duration Limit* が定義されていると, *Attempt Absolute Duration Control* が True に設定される。逆に *Limit Condition Attempt Limit* が定義されていない場合, *Limit Condition Attempt Control* は False に設定される。

5.1.3.3 Attempt Experienced Duration (試行経験経過時間)

IMS SS 仕様には, 絶対時間と経験時間の 2 種類の時間セットがある。経験時間とは学習者が該当のアクティビティに実際に対話的に試行した時間である。経験時間にはアクティビティが中断されている時間は含まれない。コンテンツの作者が *Limit Condition Attempt Experienced Duration Limit* 値を定義しない場合, アクティビティへの試行の経験時間に関しては何も制限が課せられない。

No.	名称	説明	値空間	既定値
5	<i>Limit Condition Attempt Experienced Duration Control</i>	アクティビティに対する 1 回あたりの最大試行時間が設定されていることを示す。 ゼロ値なら, アクティビティに対してはなんら制限時間が設定されていないことを示す。	論理型	False
6	<i>Limit Condition Attempt Experienced Duration Limit</i>	アクティビティに関する, 1 回あたりの学習者による最大試行経験時間。この値は学習者が, 実際に該当のアクティビティに会話的に費やした時間で, アクティビティの中断時間 (すなわち, アクティビティが経験されないまたは非アクティブな時間) は含まれない。ゼロ値は, 該当のアクティビティがアクセスできないことを示す。 <i>Limit Condition Activity Attempt Experienced Duration Control</i> 値が True でない場合, この	時間 - 単位 (精度) は 0.1 秒	0.0

		値は不定値である。		
--	--	-----------	--	--

表5.1.3.3a: Attempt Absolute Duration Limit

5.1.3.4 Activity Absolute Duration (アクティビティ絶対経過時間)

ここまでの2つの時間制限(*Attempt Absolute Duration Limit* と *Attempt Experienced Duration Limit*)は、アクティビティに関する一回ごとの試行に焦点をあてている。IMS SSの仕様にはこれ以外にも、アクティビティへの(複数回の)試行の合計時間での制限も可能になっている。Activity Absolute Durationはアクティビティに対する合計の最大時間を制限できる条件である。この時間は、ある学習者に関する、すべての *Attempt Absolute Duration* を合計することで計算される。コンテンツの作者は、アクティビティに対する学習者の合計試行時間に焦点をあてたシーケンシングルールを設定できる。

No.	名称	説明	値空間	既定値
7	<i>Limit Condition Activity Absolute Duration Control</i>	アクティビティに対する最大許容時間(の合計)が設定されていることを示す。 False 値の場合、アクティビティに関する許容時間の制限は行われない。	論理型	False
8	<i>Limit Condition Activity Absolute Duration Limit</i>	アクティビティに対して許容される(全試行の合計での)最大試行時間。ゼロ値ならアクティビティにはアクセスできないことになる。 <i>Limit Condition Activity Absolute Duration Control</i> 値が True 以外のときのこの要素値は不定値である。	時間-単位 (精度) 0.1 秒	0.0

表5.1.3.4a: Activity Absolute Duration Limit

5.1.3.5 Activity Experienced Duration (アクティビティ経験経過時間)

アクティビティ経験時間はコンテンツの作者が学習者のアクティビティに対する許容経験時間を設定するための上限を定義する。*Limit Condition Attempt Experienced Duration Limit* の場合と同じく、この制限(条件)は学習者がアクティビティを実際に(複数回)経験する時間に対してのみ制限を加える。IMSのSS仕様にはこの時間をどのように算出するかに関する要求はなく、したがってこの上限条件は信頼できないもの(とみなすべき)で、仕様には注意が必要である。

No.	名称	説明	値空間	既定値
9	<i>Limit Condition Activity Experienced Duration Control</i>	アクティビティに対して、学習者に許容する(全試行の合計での)最大試行消費時間が設定されていることを示す。 これが False 値の場合、この経験時間の制限	論理型	False

		はアクティビティに関しては行われなことを示す。		
10	<i>Limit Condition Activity Experienced Duration Limit</i>	<p>アクティビティへの学習者ごとの全消費時間の最大値での制限値。この制限は、アクティビティに対する学習者の経験時間だけに適用され、アクティビティの中断時間（すなわちアクティビティに対する非経験時間と非アクティブ時間）には適用されない。ゼロ値はこのアクティビティにはアクセスできないことを示す。</p> <p><i>Limit Condition Activity Experienced Duration Control</i> 値が True の場合以外は、この値は不定値である。</p>	時間- 単位 (精度) 0.1 秒	0.0

表 5.1.3.5a: *Activity Experienced Duration Limit*

5.1.3.6 Begin Time Limit (開始時間制限)

コンテンツの作者が、いつアクティビティを学習者に配信するかを制限したい、というシナリオを想定することができる。IMS SS の仕様では、コンテンツの作者がこのような条件を定義できるようにしている。*Begin Time Limit* は、それ以前にはアクティビティが配信不可能である日時を定義する。*Begin Time Limit* の定義がアクティビティに対して存在しない場合、このアクティビティに対して学習者がいつアクセスするかには、制限は課せられないことになる。

No.	名称	説明	値空間	既定値
11	<i>Limit Condition Begin Time Limit Control</i>	<p>アクティビティに対する利用可能条件が設定されていることを示す。</p> <p>False 値の場合、アクティビティにはアクセス制限が課されていないことになる。</p>	論理型	False
12	<i>Limit Condition Begin Time Limit</i>	<p>それ以前はアクティビティが利用不可能である日時。</p> <p><i>Limit Condition Begin Time Limit Control</i> 値が True の場合以外は、この値は不定値である。</p>	時間ポイント (Time Point) - 単位 (精度) 0.1 秒	October, 15 1582 00:00:00.0

表 5.1.3.6a: *Begin Time Limit*

5.1.3.7 End Time Limit (終了時間制限)

コンテンツの作者が、いつアクティビティを学習者に配信できなくなるかを制限したい、というシナリオを想定することができる。IMS SS の仕様では、コンテンツの作者がこのような条件を定義できるようにしている。*End Time Limit* は、それ以降にはアクティビティが配信不可能である日時を定義する。*End Time*

Limit の定義がアクティビティに対して存在しない場合，このアクティビティに対して学習者がいつアクセスするかには，制限は課せられないことになる．

No.	名称	説明	値空間	既定値
13	<i>Limit Condition End Time Limit Control</i>	アクティビティに対して利用可能時刻が設定されていることを示す． 「False」値は、アクティビティのアクセスには制限が課されていないことを示す．	論理型	False
14	<i>Limit Condition End Time Limit</i>	それ以降はアクティビティが利用不可能になる日時． <i>Limit Condition End Time Limit Control</i> 値が「True」値以外の場合、この値は不定値である．	時間ポイント (Timepoint) – 単位(精度) 0.1 秒	October, 15 1582 00:00:00.0

表 5.1.3.7a: *End Time Limit*

5.1.4. 補助リソース

アクティビティにはそれ自身に関連する補助リソースを設定でき学習者に付加サービスや付加リソースを提供できる．IMS SS の仕様ではこの補助リソースに関しては何も説明も意味付けも行われていない．どの補助リソースを利用可能とし，どのように使用するかも定義されていない．IMS SS の仕様で定義されているのはアクティビティに補助リソースを関連付ける手段だけである．

IMS SS の仕様では，与えられたアクティビティに補助リソースを関連付ける要素だけが提供されてる．*Resource ID* (値空間は固有識別子)と *Purpose* (値空間は任意の語彙(open category))である．*Resource ID* で補助リソースが識別され，*Purpose* で識別された補助リソースの目的が示される．

IMS SS の仕様による定義された要件は以下のとおりである：

1. アクティビティが学習者に配信されるとき，補助リソースも学習者に利用可能であること．
2. 補助リソースは必ずしも存在しなくてもよく，各アクティビティに定義されていなくてもよいこと．
3. 補助リソースがアクティビティに配信される場合，固有の識別子と目的が提供されること．

IMS SS の仕様では，*Purpose* 要素はオープン，(言い換えると)未定義の語彙と記述されている．また既定の補助リソースも(目的に与えられる)トークン値も記述されていない．

5.1.5. ロールアップルールについて

IMS SS の仕様では、アクティビティに関するロールアップの動作を個々のルールごとに詳細に定義している。ロールアップとは、子アクティビティの集合に対する学習目標の進捗情報と試行に関する進捗情報を評価したうえで親のそれら（学習目標の進捗情報と試行の進捗情報）を決定するプロセス、と定義される。

ゼロ個以上のロールアップルールの集合をアクティビティに対して定義できる。ロールアップルールはロールアッププロセス中に評価される（7.1.5 節ロールアップ動作参照）。ロールアップルールの評価により、既定のシーケンシングパスが変更されたり、アクティビティおよびこれらの子アクティビティティを利用可能性に影響をおよぼすことがある。

ロールアップルールは、対象となる子アクティビティの集合の定義、これら子アクティビティに対するトラッキング情報を評価する条件の集合、および、これらの条件の集合が True に評価されるときに取られる該当のアクションや動作 (*if [child_activity_set condition_set] then [action/behavior]*) から構成される。ロールアップルールは省略可能でコンテンツの作者が必要になったときだけ定義するものである。おのこのルール属性には規定値が定義されている。データが実体化されていない場合は既定値が適用される。

5.1.5.1 Child Activity Set (子アクティビティ集合)

Rollup Child Activity Set はロールアップルールの処理中に使用される子の集合を定義する。IMS SS の仕様で定義されたいくつかのメカニズムによって、コンテンツの作者は評価時に使用される子アクティビティを定義できるようになっている。Child Activity Set は評価のプロセスで対象とされる子アクティビティの集合を記述する固定語彙の集合から成る。

- All (既定値)：親のアクティビティに属するすべての子の組み合わせが True と評価されるロールアップ条件の集合をもつなら、指定されたアクションがとられる。
- Any：親のアクティビティに属する子のいずれかが True と評価されるロールアップ条件の集合をもつなら、指定されたアクションがとられる。
- None：親のアクティビティに属する子のうち True と評価されるロールアップ条件の集合をもつものがないなら、指定されたアクションがとられる。
- At Least Count：少なくとも、親のアクティビティの *Rollup Minimum Count* により示される個数の子が、True と評価されるロールアップ条件の集合をもつなら、指定されたアクションがとられる。
- At Least Percent：少なくとも、親のアクティビティの *Rollup Minimum Percent* により示される（百分率の）割合の個数の子が、True と評価され

るロールアップ条件の集合をもつなら，指定されたアクションがとられる．

ロールアップルール条件の記述に At Least Count の語彙が使用されるとき，*Rollup Minimum Count* の値が使用される．この値は，ロールアップ条件の集合が True でなくてはならない子アクティビティの最小数を示す整数である．*Rollup Minimum Count* の既定値は 0 である．したがってこの値を指定しないとロールアップ評価時に，どの子も考慮されないことになり，アクションが無条件で実行されることになる．

ロールアップルール条件の記述に At Least Percent の語彙が使用されるときには，*Rollup Minimum Percent* の値が使用される．この値は 0.0 から 1.0 のあいだの 0.0 と 1.0 を含む実数である．この値はロールアップ条件の集合が True でなくてはならない子アクティビティの最小の個数の割合を指定する．*Rollup Minimum Percentage* の既定値は 0.0 である．したがってこの値を指定しないとロールアップ評価時に，どの子も考慮されないことになり，アクションが無条件で実行されることになる．

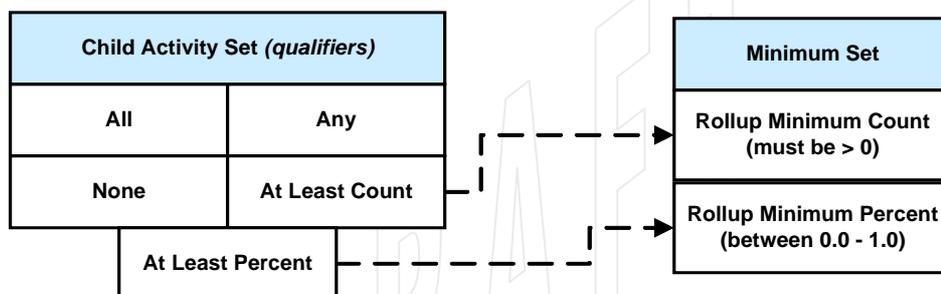


図 5.1.5.1a: Minimum Set Relationship

IMS SS の仕様では柔軟性が組み込まれていて，さまざまな異なる状況に応じてさまざまなロールアップルールが定義されている．以下の図は，記述されたさまざまな要素を使用してロールアップルールがどのように構築されるかを，簡単な 2 例で示している：

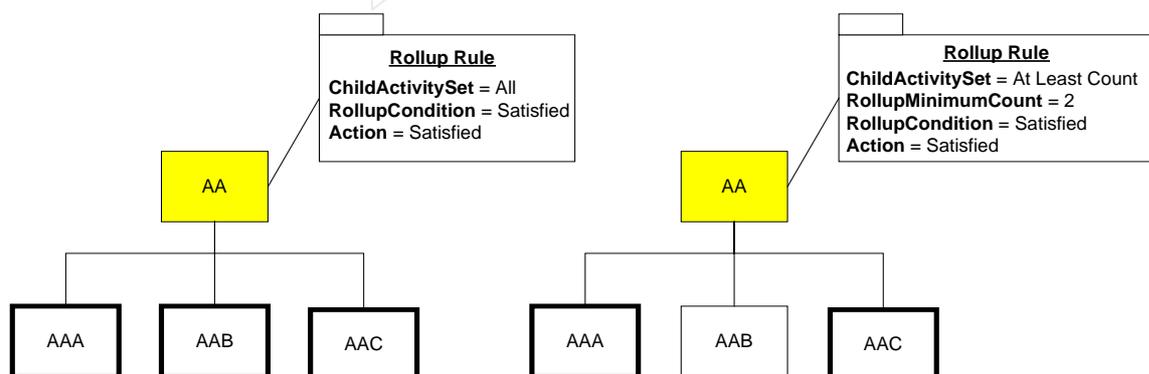


図 5.1.5.1b: Rollup Rule Illustration

最初のシナリオのルールではアクティビティ AA の All (すべて) の子が “Satisfied” (Rollup Condition) なら , アクティビティ AA も “Satisfied” (Action) とみなされるべき条件を定義している . 二番目のシナリオではアクティビティ AA の少なくとも 2 つの子が “Satisfied” (Rollup Condition) なら , アクティビティ AA も “Satisfied” (Action) とみなされるべきことを示している . こういったロールアップルールと該当のアクションをベースにして , アクティビティ AA が “Satisfied” なら “Skip” するなどのシーケンシングルールをアクティビティ AA に適用できる .

5.1.5.2 Rollup Condition Combination (ロールアップルールの組み合わせ)

ロールアップルールを組み合わせると , ルール集合を作成できる . これらのルール集合を修飾して , LMS にどのルールを評価するかを指示できる . IMS SS の仕様では , 2 タイプの修飾子 (Condition Combination) を定義している .

- All : ルール条件集合は , 個々のロールアップ条件の全てが True のとき , True と評価される . 論理積 (logical and) として働く .
- Any (既定値) : ルール条件集合は , 個々のロールアップ条件のいずれかが True のとき , True と評価される . 論理和 (logical or) として働く .

5.1.5.3 Rollup Conditions (ロールアップ条件)

IMS SS の仕様では , コンテンツの作者は Rollup Conditions の集合を定義できるようになっている . Rollup Conditions 要素は , ロールアップルール集合が定義されたアクティビティのコンテキストで評価される単一の条件からなる Rollup Condition ステートメントの集合である . したがってコンテンツの開発者は Condition Combination 要素を使用して LMS が条件の集合をどのように評価するかを記述できることになる . Condition 要素は , IMS SS 仕様のトラッキングモデルをベースとする制限語彙の集合からなる :

条件	説明
Satisfied	子のアクティビティに関連するロールアップ学習目標に対する <i>Objective Progress Status</i> が True , かつ子のアクティビティに関連するロールアップ学習目標に対する <i>Objective Satisfied Status</i> が True ならこの条件は True に評価される .
Objective Status Known	子のアクティビティに関連するロールアップ学習目標に対する <i>Objective Progress Status</i> が True ならこの条件は True に評価される .
Objective Measure Known	アクティビティに関連するロールアップ学習目標に対する <i>Objective Measure Status</i> が True ならこの条件は True に評価される .
Completed	子のアクティビティに対する <i>Activity Progress Status</i> が True , かつ子のアクティビティに対する <i>Attempt Completion Status</i> が True ならこの条件は True に評価される .
Activity Progress Known	<i>Activity Progress Status</i> が True , かつ子のアクティビティに対する <i>Attempt Progress Status</i> が True ならこの条件は True に評価される .

Attempted	子のアクティビティに対する <i>Activity Progress Status</i> が True , かつ子のアクティビティに対する <i>Activity Attempt Count</i> が正 (すなわち子のアクティビティが試行されている) ならこの条件は True に評価される .
Attempt Limit Exceeded	子のアクティビティに対する <i>Activity Progress Status</i> が True , かつ子のアクティビティに対する <i>Limit Condition Attempt Control</i> が True , なおかつ子のアクティビティに対する <i>Activity Attempt Count</i> が子のアクティビティに対する <i>Limit Condition Attempt Limit</i> に等しいか , これより大きいならこの条件は True に評価される .
Time Limit Exceeded	子のアクティビティに対する <i>Activity Progress Status</i> が True , かつ子のアクティビティに対する記録された時間値のいずれか : <ul style="list-style-type: none"> • <i>Activity Absolute Duration</i>, • <i>Activity Experienced Duration</i>, <i>Activity</i> • <i>Attempt Absolute Duration</i>, <i>Activity</i> • <i>Attempt Experienced Duration</i> が , 子のアクティビティに対する , 以下の該当 の時間制限条件 (<i>Limit Condition</i>) 値 : <ul style="list-style-type: none"> ▪ <i>Limit Condition Activity Absolute Duration Limit</i>, ▪ <i>Limit Condition Activity Experienced Duration Limit</i>, <i>Activity</i> ▪ <i>Limit Condition Attempt Absolute Duration Limit</i>, <i>Activity</i> ▪ <i>Attempt Experienced Duration Limit</i> を超えていて , かつ以下の関連する制限条件コントロール値 : <ul style="list-style-type: none"> ▪ <i>Limit Condition Activity Absolute Duration Control</i>, ▪ <i>Limit Condition Activity Experienced Duration Control</i>, ▪ <i>Limit Condition Attempt Absolute Duration Control</i>, ▪ <i>Limit Condition Attempt Experienced Duration Control</i> が True のとき , この条件は True に評価される .
Outside Available Time Range	これを評価される際の時刻 (<i>current time</i>) が , 子のアクティビティに対する該当の時刻制限条件の前または過ぎている : <ul style="list-style-type: none"> ▪ <i>Limit Condition Begin Time Limit</i>, ▪ <i>Limit Condition End Time Limit</i> , かつ子のアクティビティの , 関連する制限条件コントロール値 : <ul style="list-style-type: none"> ▪ <i>Limit Condition Begin Time Control</i>, ▪ <i>Limit Condition End Time Control</i> が True ならこの条件は True に評価される .

表 5.1.5.3a: Rollup Conditions

5.1.5.4 Rollup Condition Operator (ロールアップ条件演算子)

Rollup Condition Operator 要素は個々のロールアップ条件に適用される単項論理演算子である . 現在の IMS SS の仕様では , この要素に 2 つの値を定義している :

- NO-OP (既定値) : ルール条件がそのまま使用される
- Not : ルール評価前にルール条件が逆転される

5.1.5.5 Rollup Actions (ロールアップ動作)

Rollup Action は、ロールアップルールを含むアクティビティに（単一または一連のルールの評価の結果が True なら）適用したいアクションを記述する。ロールアップされた結果の動作は、関連するアクティビティの Objective Progress Status または Activity Attempt Progress に適用される：

ロールアップアクション	アクションの説明
Satisfied (既定値)	<ul style="list-style-type: none"> ▪ アクティビティに関するロールアップ学習目標に対する <i>Objective Progress Status</i> を True に設定し、 ▪ アクティビティに関するロールアップ学習目標に対する <i>Objective Satisfied Status</i> を True に設定する。
Not Satisfied	<ul style="list-style-type: none"> ▪ アクティビティに関するロールアップ学習目標に対する <i>Objective Progress Status</i> を True に設定し、 ▪ アクティビティに関するロールアップ学習目標に対する <i>Objective Satisfied Status</i> を False に設定する。
Completed	<ul style="list-style-type: none"> ▪ アクティビティに対する <i>Attempt Progress Status</i> を True に設定し、 ▪ アクティビティに対する <i>Attempt Completion Status</i> を True に設定する。
Incomplete	<ul style="list-style-type: none"> ▪ アクティビティに対する <i>Attempt Progress Status</i> を True に設定し、 ▪ アクティビティに対する <i>Attempt Completion Status</i> を False に設定する。

表5.1.5.5a: Rollup Actions

5.1.5.6 ロールアップルールの例

図 5.1.5.6a に、学習アクティビティに対するロールアップルールの定義例を示す。

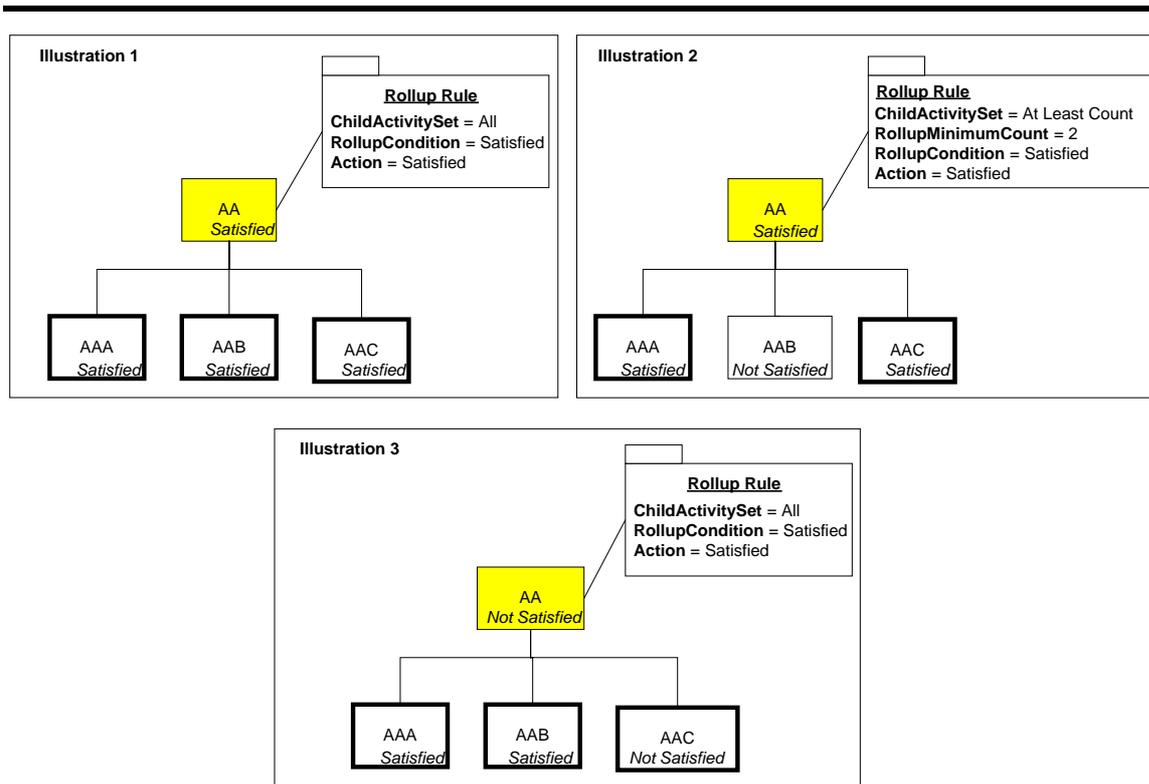


図5.1.5.6a: Rollup Rule Condition Illustration

上図中の図 1 には、親 (AA) アクティビティが習得済みとみなされるためには、すべての子アクティビティ (AAA, AAB および AAC) が習得済みでなければならないという意味のロールアップルールが示されている。図 2 には、親 (AA) アクティビティが習得済みとみなされるためには、子アクティビティのうち 2 つ以上が習得済みでなければならないという意味のロールアップルールが示されている。図 3 には成立しなかった (習得済みではない) ロールアップルールが示されている。このロールアップルールは、親 (AA) アクティビティが習得済みとみなされるためには、すべての子アクティビティ (AAA, AAB および AAC) が習得済みでなければならないという意味である。子アクティビティ AAC の状態は習得済みでないので、ロールアップルールにより親の AA は非習得であるとみなされる。

5.1.6. ロールアップコントロール

IMS SS 仕様では、コンテンツの作者によって、あるアクティビティがロールアップ対して供給する情報を条件付きで制限できるようになっている。ロールアップコントロールには、アクティビティに対して指定されたロールアップ動作のタイプが記述されている。現在、3 タイプのトラッキングモデルの情報がロールアッププロセス中に使用できる：

- 学習目標の習得状態 (*Objective Satisfied Status*)

- 学習目標の習得度(*Objective Normalized Measure*)
- アクティビティ完了状態(*Attempt Completed Status*)

最初の2つの値は該当の学習目標がロールアップに関係する(*Objective Contributes to Rollup* が True 値のとき)場合に限ってロールアップに適用される。

No.	名称	説明	値空間	既定値
1	<i>Rollup Objective Satisfied</i>	アクティビティに関する学習目標 (<i>Objective Contributes to Rollup</i> が True 値をもつ) に対する <i>Objective Satisfied Status</i> 値が親のアクティビティに対するロールアップに含まれている (True または False) かどうかを示す。	論理型	True
2	<i>Rollup Objective Measure Weight</i>	親のアクティビティへのロールアップ中に使用される, (子)アクティビティに関する学習目標に対する <i>Objective Normalized Measure</i> (<i>Objective Contributes to Rollup</i> が True のもの) に適用される重み付けファクター。	実数[0..1] 精度は最低 4 桁の十進数	1.0
3	<i>Rollup Progress Completion</i>	アクティビティに対する <i>Attempt Completed Status</i> 値が親のアクティビティへのロールアップに含まれるかどうか (True/False かによる) を示す。	論理型	True

表5.1.6a: *Rollup Controls*

5.1.7. 学習目標

IMS SS 仕様を SCORM に導入することにより, 学習目標をアクティビティに関連付けするメカニズムが設けられた。各学習アクティビティにはいくつでも学習目標を定義することができる。トラッキングモデルを使用して, あるアクティビティの一回ごとの試行に対して定義されている各学習目標ごとの習得状態 (“習得” または “非習得”) と習得度 (たとえば得点) を記録するのに使用するデータセットを定義する。IMS SS の仕様によるメカニズムは, 学習アクティビティを ID に関連付けするものである。学習目標の意味づけ自体は定義されていない。

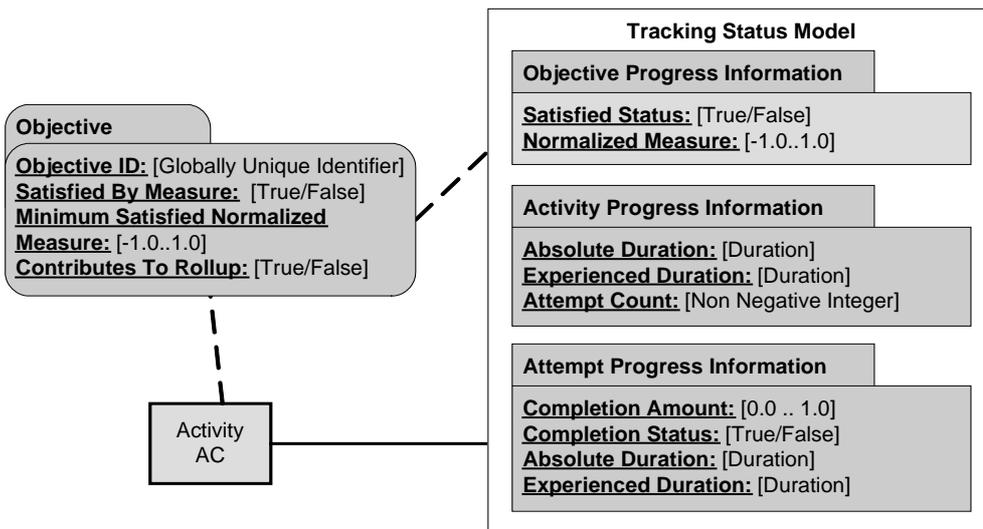


図 5.1.7a: 学習目標説明(Objective Description)と学習目標進捗情報(Objective Progress Information)の関係

各学習目標には一連の情報を含めるように定義できる。IMS SS の仕様では、各学習目標を記述するのに以下の要素を定義している：

- **学習目標 ID**： アクティビティに関係付けられた学習目標に対するグローバルで固有の識別子。ID は対応する学習目標情報とアクティビティ間のリンクとして働く。学習目標 ID には既定値はない。ひとつのアクティビティに対してひとつ以上の学習目標が定義されるときにだけ、学習目標 ID が必要となる。
- **Objective Satisfied By Measure**（既定値は *False*）： 学習目標の満足度を測定値（たとえば得点）によって決定するかどうかを示す。この値は、アクティビティに関する学習目標が満足されたかどうかを決定する他の方法(トラッキングモデル – *Objective Satisfied Status*)の代わりに、*Objective Minimum Satisfied Normalized Measure* を使用するかどうかを示す。
- **Objective Minimum Satisfied Normalized Measure**（既定値は *1.0*）： 学習目標に対する最小習得度。この習得度は、-1.0 と 1.0 の間で-1.0 と 1.0 を含むように正規化する必要がある。 *Objective Satisfied by Measure* が *True* に設定されていると：
 - 学習目標に対する *Objective Measure Status* が *True* で同じく学習目標に対する *Objective Normalized Measure* がこの値を超えていると、*Objective Progress Status* が *True* に設定され *Objective Satisfied Status* が *True* に設定される。
 - 学習目標に対する *Objective Measure Status* が *True* で学習目標に対する *Objective Normalized Measure* がこの値より小さければ、*Objective Progress Status* が *True* に設定され *Objective Satisfied Status* が *False* に設定される。
 - 学習目標に対する *Objective Measure Status* が *False* なら、*Objective Progress Status* は *False* に設定される。

ADL よりの注意 : この要素の導入により , SCORM1.2 での MasteryScore 要素は廃止される . SCORM1.3 のマニフェストでは MasteryScore 要素は使用してはならない . この要素は現在の SCORM1.2 の MasteryScore を置き換えるものであるから , 現在の SCORM1.2 での要素に対するマッピングは依然として存在する . ロールアップに貢献する学習目標 (プライマリ学習目標) に対する *Objective Minimum Satisfied Normalized Measure* value 値は , `cmi.student_data.mastery_score` の初期化に使用しなければならない . もし *Objective Satisfied by Measure* が True のプライマリ学習目標が何もないなら , `cmi.student_data.mastery_score` は 0 とみなさなければならない .

- **Objective Contributes to Rollup** (既定値は *False*) : 学習目標に対する *Objective Normalized Measure* および *Objective Satisfied Status* をロールアップ評価時に使用するかどうかを示す .

複数の学習目標がひとつのアクティビティに定義されている場合 , 少なくともひとつの主学習目標が指定されていなければならない . 主学習目標とはロールアップ時に使用される学習目標である .

5.1.7.1 ローカル学習目標とグローバル学習目標

学習目標は , 学習アクティビティとは別のものとみなされる . 学習目標はローカルまたはグローバルの有効範囲を持つデータ項目の集合で , おのおの習得状態と習得値をもつものとして表現される . アクティビティはひとつ以上のローカルな学習目標を持つことができ , 複数のグローバル共有学習目標を参照できる . 複数のアクティビティは同じグローバル学習目標を参照してもかまわないので , この場合はデータ値は共有されることになる . これによって , あるアクティビティが他のアクティビティの状態に影響する , というような複雑なシーケンシングルールも設定可能である .

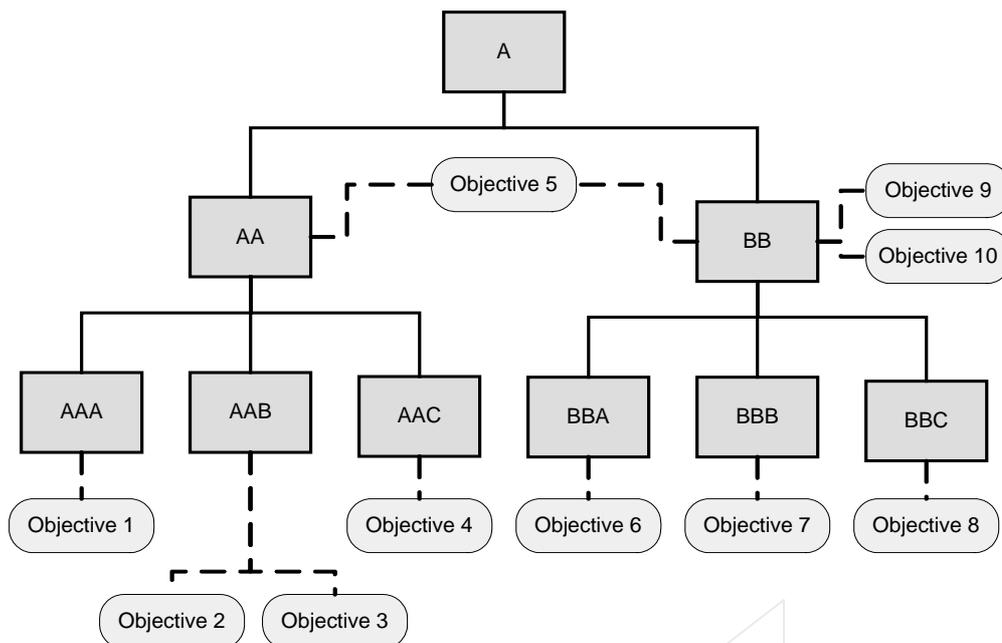


Figure 5.1.7.1a: 学習目標の共有

ローカル学習目標 ID (objective ID) とグローバル学習目標 ID (objective ID) の指定方法は決められていない。ある学習目標は単一のアクティビティツリー内で共有可能だし、複数のアクティビティのツリーでグローバルに共有されていてもかまわない。アクティビティは複数の学習目標を参照できるので、アクティビティが副学習目標 (群) を持つようなメカニズムが提供できる。

5.1.7.2 学習目標の共有 – 学習目標マッピング

Objective Map の記述では、ひとつの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の間のマッピングを定義する。学習目標のマッピングによって、アクティビティ間の共有情報をどのように利用するかが決定される。このマッピングに適用されるときルールは以下のとおりである：

- 各アクティビティは無制限個の学習目標マップを持つことができる。
- 既定値では、アクティビティ間で共有されるグローバル学習目標はない。ローカル学習目標情報をグローバル共有学習目標にマップする際には、*Objective Maps* の集合を定義して、その方法を規定する必要がある。
- *Objective Map* データはローカルの学習目標情報が変更されるときは必ず、トラッキングモデルの動作で記述されたとおりに評価される。
- どのようなローカル学習目標に対しても、最大でひとつのグローバル学習目標からの「読み出し」マッピングを定義できる。
- どのようなグローバル学習目標に対しても、任意のアクティビティにおいて最大でひとつのローカル学習目標からの「書き込み」マッピングを定義できる。

5.1.7.2.1 共有学習目標の読み出し / 書き込み

学習の過程であるアクティビティの学習目標の状態を他のアクティビティに反映させるルールを、コンテンツの作者が定義したいことがある。たとえば、特定のアクティビティの学習目標が達成（満足）されると他の一連のアクティビティを学習者から見えなくしたい（事前テストに合格すると事前テストに関する学習目標に関する教材を見る必要がなくなる、など）場合がある。これ以外にも同じようなシナリオが考えられるであろう。このようなタイプのシナリオをコンテンツの作者に利用可能にするために、どのグローバルデータを、どのアクティビティで共有させるかを定義できるようになっている。Objective Map のメカニズムは、IMS SS の仕様に定義されたもので、上記のようなシナリオを利用可能にしている。

こういった共有学習目標に対して 2 種類のアクセス方法がある：*reading from* と *writing to* である。既定値では、各アクティビティは少なくともひとつの定義されたローカル学習目標を持っているので、いつ共有学習目標へアクセスするかの条件が定義されていなくてはならない。学習目標情報が必要でローカル学習目標が未定義のときはいつでも、共有学習目標は「読み出」されなくてはならない。各アクティビティのローカル学習目標の情報は共有学習目標のそれらより優先される。したがってもしルールが評価された結果、ローカル学習目標情報によって学習目標が満足されたことが示されると、グローバル学習目標の情報にはアクセスが行われてはならない。しかし、ローカル学習目標の学習目標情報が未定義で、かつ学習目標マップにより習得状態の読み出しルールが定義されている場合（True の場合）、共有グローバル学習目標を読み出さなくてはならない。

アクティビティに対する学習目標が共有グローバル学習目標への書き込み定義を持つ（*Write Objective Satisfied Status* が True）場合、アクティビティに関する学習目標状態の情報がアクティビティの終了時に、学習目標 ID（*Target Objective ID*）によって識別されるグローバル学習目標に転送される。

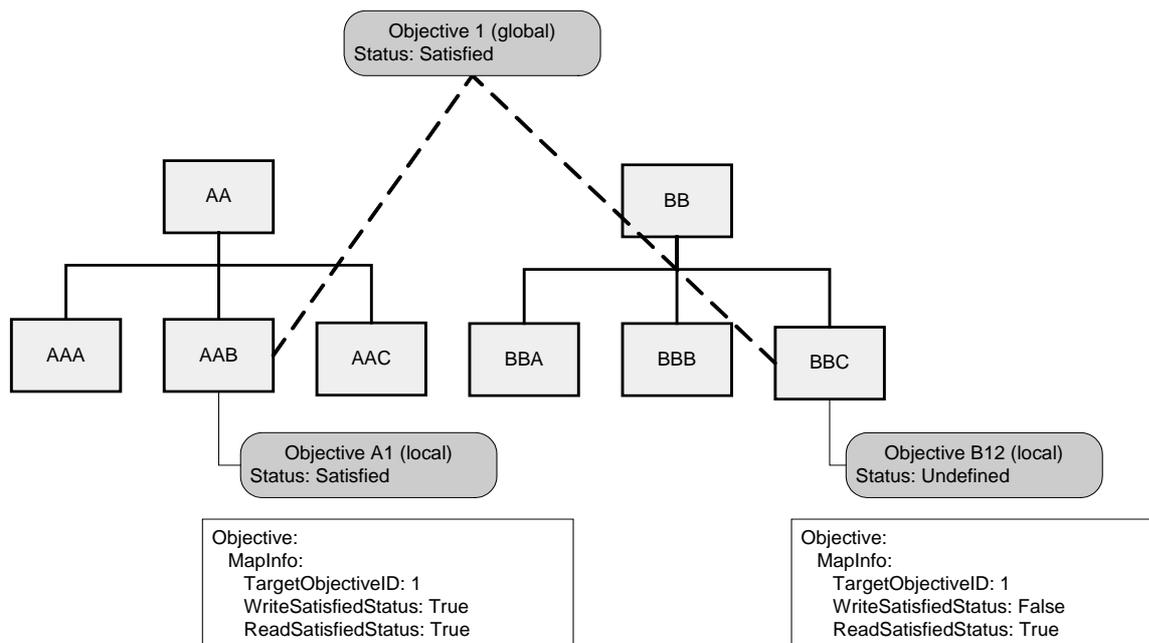


図5.1.7.2.1a: 共有学習目標

図 5.1.7.2.1a では、学習目標 1 はグローバル学習目標に指定されている。アクティビティ AAB と BBC はどちらもこのグローバル学習目標を共有している（おのこの TargetObjectiveID 属性で示される）。アクティビティ AAB に対する学習目標情報により、学習目標状態の読み出しが必要な場合、グローバル学習目標（学習目標 1）から行われ、同じく書き込みも行われる。したがってアクティビティ AAB が終了するときに、学習目標状態は LMS によってグローバル学習目標（学習目標 1）に転送される。シーケンシングルールを処理して次に配信するものを決定する際、LMS はアクティビティ BBC の学習目標状態にアクセスすることがある。BBC の学習目標状態は未定義状態のため LMS はこのアクティビティに対して定義されている学習目標情報をチェックしなければならない。この学習目標情報に従い、学習目標状態が未定義であれば LMS は該当のグローバル学習目標を読み出す。このケースでは LMS は、Satisfied となっているグローバル学習目標（学習目標 1）を読み出す。これによってアクティビティ BBC も Satisfied となり、これに基づいてルールが解釈される。

No.	名称	説明	値空間	既定値
1	Activity Objective ID	アクティビティに関連するローカル学習目標の識別子。 注：Activity Objective ID に対する既定値はない。もしアクティビティに対して学習目標マップが定義されているなら、Activity Objective ID には値が必要になる。	固有の識別子	なし 値は必須
2	Target Objective ID	マッピングの対象となるグローバル共有学習目標の識別子。 注：Target Objective ID に対する既定値はない。もしアクティビティに対して学習目標マップ	固有の識別子	なし 値は必須

		ブが定義されているなら，Target Objective ID には値が必要になる．		
3	<i>Read Objective Satisfied Status</i>	ローカル学習目標の進捗が未定義のとき (<i>Activity Objective ID</i> で識別されるローカル学習目標の <i>Objective Progress Status</i> が False) ， (<i>Activity Objective ID</i> により識別される) ローカル学習目標の <i>Objective Satisfied Status</i> を該当 (<i>Target Objective ID</i> により識別される) のグローバル学習目標から読み出すかどうか (True または False により) を示す． この操作は，ローカル学習目標に関連する学習目標情報を変更しない．	論理型	True
4	<i>Write Objective Satisfied Status</i>	アクティビティの終了時に，これに関連する (<i>Activity Objective ID</i> により識別される) ローカル学習目標の <i>Objective Satisfied Status</i> を (<i>Target Objective ID</i> により識別される) 該当のグローバル共有学習目標に移送するかどうか (True または False により) を示す．	論理型	False
5	<i>Read Objective Normalized Measure</i>	ローカル学習目標の測定値が未定義のとき (<i>Activity Objective ID</i> により識別されるローカル学習目標に対する <i>Object Measure Status</i> が False) ，アクティビティに関連する学習目標により識別される (<i>Activity Objective ID</i> による) <i>Objective Normalized Measure</i> を該当 (<i>Target Objective ID</i> により識別される) のグローバル学習目標から読み出すかどうか (True または False により) を示す． この操作は，ローカル学習目標に関連する学習目標情報を変更しない．	論理型	True
6	<i>Write Objective Normalized Measure</i>	アクティビティの終了時に，これに関連する (<i>Activity Objective ID</i> により識別される) 学習目標の <i>Objective Normalized Measure</i> 値を (<i>Target Objective ID</i> により識別される) 該当のグローバル共有学習目標に移送するかどうか (True または False により) を示す．	論理型	False

図 5.1.7.2.1a: グローバル学習目標コントロールの読み出し/書き込み

5.1.7.2.2 グローバル学習目標の寿命 (有効期間)

グローバル学習目標の寿命 (有効期間) について疑問がわくであろう．最小期間として考えると，あるアクティビティに対するグローバル学習目標 (ならびに関連する情報) がある学習者用に，いったん作成されると，この学習者が LMS を使用している間，このグローバル学習目標 (ならびに関連する情報) は有効とされる．このことから，グローバル学習目標は，オリジナルのアクティビティの範囲外となるような他のアクティビティと共有させることも可能である (たとえば LMS 下で複数のコースを使用した学習など) ．

5.1.8. アクティビティの選択の制御

コンテンツの作者は、特定のアクティビティがいつ選択されるか、またアクティビティの選択回数の上限を設定するシーケンシングルールを定義できる。つまり、LMSのシーケンシングエンジンに「あるアクティビティに対する最初の試行時に、6つある(子)アクティビティの中から4つをピックアップせよ」と指示するルールを書くことも可能である。

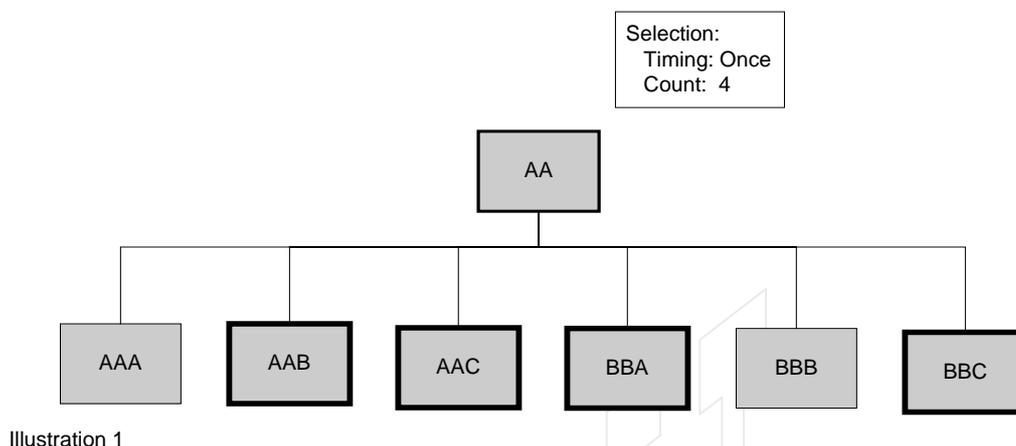


Illustration 1

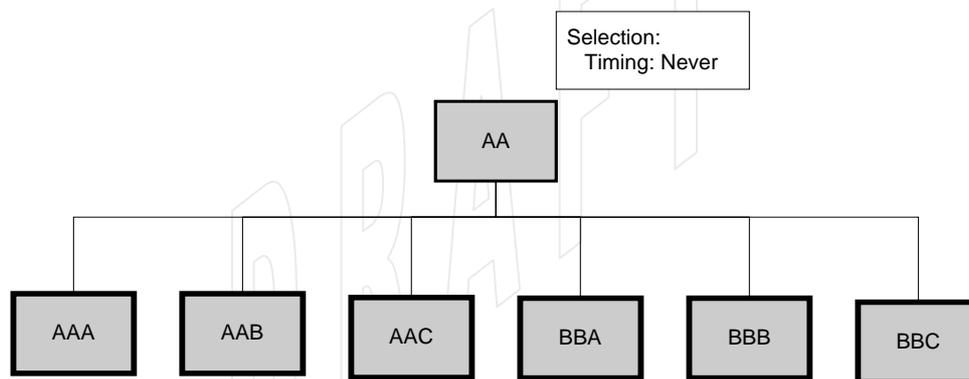


Illustration 2 (default case)

図5.1.8a: 選択の例

図1では、アクティビティ AA への最初の試行時に、LMSのシーケンシングエンジンは6つの子の中から4つを配信することを示している。もし学習者がアクティビティ AA への試行を中断しても、この情報は保持されなければならない。選択カウントが子の数より大きければ全アクティビティが配信の選択対象候補になる。

図2では、LMSのシーケンシングエンジンが、学習者への配信候補に全ての子を対象にすることを示している。これはまた、選択ルールが何も指定されない場合の既定値の動作でもある。

シーケンシングのプロセス中に、あるアクティビティの子をどのように選択するかの記事も、選択コントロールに含まれている。子アクティビティの選択は「新規に試行がなされる都度」に行われるのが望ましいが、この操作が行われ

るのはいつかについてはあまりよく定義されてはいない。現時点での IMS SS の仕様では、この属性の動作に関する指定はない。現時点では「新規に試行がなされる都度」での *Selection Timing* は適用するべきではない。

No.	名称	説明	値空間	既定値
1	<i>Selection Timing</i>	いつ選択が行われるかを示す。 <ul style="list-style-type: none"> • <i>Never</i> – 選択は適用されない；既定値によりアクティビティに属するすべての子が選択対象とされる。 • <i>Once</i> – アクティビティに対する最初の試行時の前に選択が適用される。 • <i>On Each New Attempt</i> – アクティビティに対する新しい試行がなされる都度選択が適用される。 現時点でのシンプルシーケンシングの仕様では、 <i>On Each New Attempt</i> のオプションと関連する動作は指定されていない。	語彙	Never
2	<i>Selection Count Status</i>	アクティビティに対する選択回数データが意味を持つかどうかを示す(True または False により決定)。	論理型	False
3	<i>Selection Count</i>	子アクティビティの数は、親のアクティビティに属する子アクティビティ集団から選択されることを示す。 <i>Selection Count</i> が子アクティビティの数よりも大きい場合、すべての子が選択される。 <i>Selection Count Status</i> が True 値でない限り、この値は信頼できない値(不定値)である。 <i>Selection Count Status</i> が False 値なら、すべての子アクティビティが選択される。	非負の整数	0

表5.1.8a: 選択コントロール

5.1.9. ランダム化コントロール

コンテンツの作者は、LMS の実装シーケンシングエンジンが、配信時に選択するアクティビティをランダム化するかどうかのシーケンシングルールを定義できる。これにより、ひとつのクラスタ内の親のアクティビティに対して最初の試行時に、どの子のアクティビティをどのようなランダム化の順で選択するかのルールを統制できる。ランダム化が行われるタイミングは以下の通りである：

- Never (既定値): ランダム化は子アクティビティには適用されない
- Once: アクティビティへの最初の試行時にのみランダム化が行われる
- On Each New Attempt: アクティビティが新たに試行されるたびにランダム化が行われる

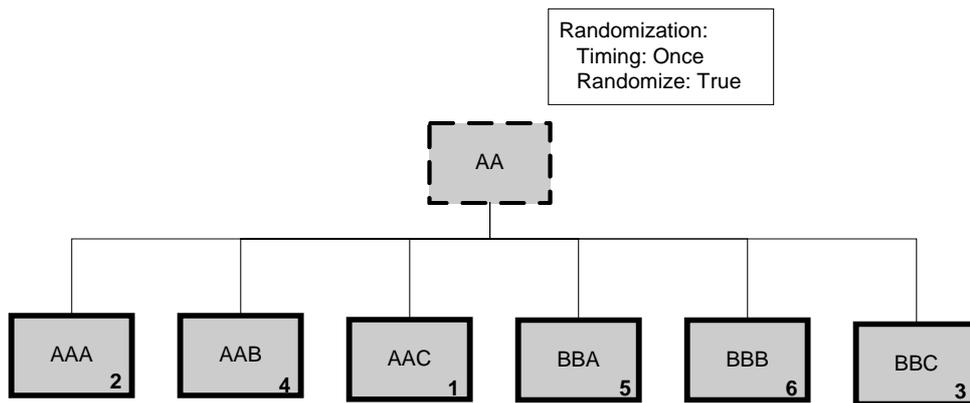


Illustration 1

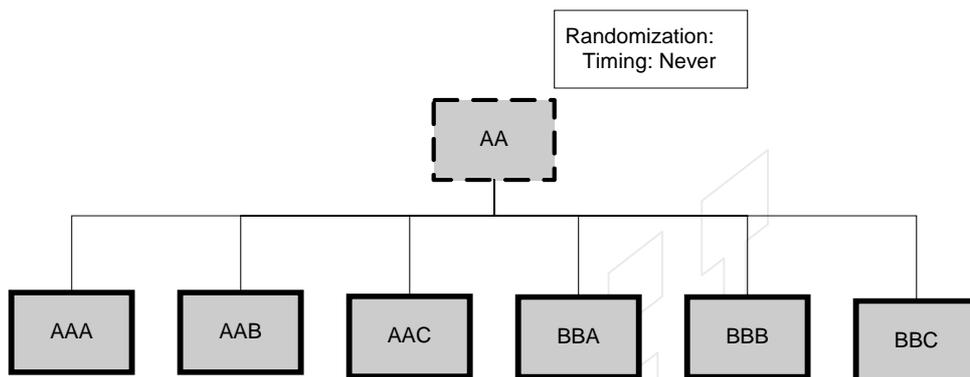


Illustration 2 (default case)

図5.1.9a: ランダム化の例

上図はランダム化ルールにおける2つの例を示している。図1では、アクティビティ AA が最初に試行されるときにランダム化を適用するように LMS に指示設定したルール例である。ランダム順の決定は LMS に任されている（ここでは各アクティビティに振られている数字で示している）。この情報は、学習者がアクティビティ AA で学習を停止した場合に備えて、LMS により保持される必要がある。図2（既定値の場合）は、LMS に子のアクティビティにランダム化を適用しないよう指示した定義例である。

No.	名称	説明	値空間	既定値
1	<i>Randomization Timing</i>	アクティビティの子の実行順を指示する： <ul style="list-style-type: none"> • <i>Never</i> – 適用しない。 • <i>Once</i> – アクティビティへの最初の試行前にだけ適用する。 • <i>On Each New Attempt</i> – アクティビティが新たに試行されるときに、常にランダム化が適用される。 	語彙	Never
2	<i>Randomize Children</i>	子アクティビティの順序がランダム化されるように指示する。(True または False)	論理型	False

表5.1.9a: Randomization Controls

5.1.10. 配信コントロール

配信コントロールは、アクティビティが配信される時の動作と制御を記述するために用いられる。これらのコントロールは、アクティビティが配信される時アクティビティ進捗データと試行進捗データを記録するかどうかを示す。コンテンツの作者は、これらのルールを設定することによってどのアクティビティに関する（進捗などの）情報を記録するか制限できる。

No.	名称	説明	種類	既定値
1	<i>Tracked</i>	アクティビティに対する学習目標進捗および試行進捗情報（Objective and Attempt Progress information）を LMS で記録，活動追跡するかどうか，またこの情報が親アクティビティのロールアップに適用するか否かを示す(True または False) . データの追跡方法と記録方法は指定されていない。	論理型	True
2	<i>Completion Set by Content</i>	アクティビティに対する <i>Attempt Completion Status</i> は，コンテンツ（オブジェクト）により（True または False に応じて）設定されることを示す。この値が False なら進捗データの設定に既定値が使用される。True の場合，完了データの設定の方法，条件，タイミングは指定されない。	論理型	False
3	<i>Objective Set by Content</i>	<i>Objective Contributes to Rollup</i> 値に True をもつ学習目標に関する <i>Objective Satisfied Status</i> は，コンテンツ（オブジェクトにより）(True または False に応じて)設定されることを示す。この値が False なら学習目標情報の設定に既定値が使用される。True の場合，学習目標情報の設定の方法，条件，タイミングは指定されない。	論理型	False

表5.1.10a: 配信コントロール

LMS は、アクティビティに関連するデータの追跡記録を行うのにこの配信コントロールを使用する。このコントロールは、アクティビティに関係するコンテンツ（SCO）が（試行進捗と学習目標進捗）データに対して特定の追跡記録を行うかどうかを示す。

5.1.10.1 Tracked（記録取得）

Tracked 要素は、アクティビティに対する学習目標進捗および試行進捗情報（Objective and Attempt Progress information）を LMS で記録，活動追跡するかどうかを示す。コンテンツの作者が指定していないときの既定値は「記録する（True）」である。LMS による記録をさせたくない場合，コンテンツの作者はこの要素を False に設定してこの要求を明示しなくてはならない。

5.1.10.2 Completion Set by Content (コンテンツによる完了設定)

Completion Set by Content は、アクティビティが完了したかどうかをコンテンツが明示する責任があることを示す。この要素に対する既定値は False である。コンテンツ (SCO) は LMS で制御される学習目標進捗および試行進捗情報 (Objective and Attempt Progress information) への直接のアクセスは有しない。SCO は、SCORM ランタイム環境用データモデル要素に該当の設定値をおくことでこれを行うことになる。学習目標進捗および試行進捗情報 (Objective and Attempt Progress information) の各要素へのマッピングについては、本書のトラッキングデータモデルの章に記述してある。この要素が True に設定され、SCO が完了進捗を設定しないときは、LMS は既定値により試行進捗情報を決定しなくてはならない。逆にこの要素が False (既定値) の場合、SCO での完了進捗状態が使用される。SCO が完了進捗状態を示していないなら、このアクティビティは完了と想定される。

5.1.10.3 Objective Set by Content (コンテンツによる学習目標設定)

Objective Set by Content は、学習目標が習得されたかどうかをコンテンツが明示する責任があることを示す。この要素の既定値は False である。コンテンツ (SCO) は LMS で制御される学習目標進捗および試行進捗情報 (Objective and Attempt Progress information) への直接のアクセスは有しない。SCO は、SCORM ランタイム環境用データモデル要素に該当の設定値をおくことでこれを行うことになる。学習目標進捗および試行進捗情報 (Objective and Attempt Progress information) の各要素へのマッピングについては、本書のトラッキングデータモデルの章に記述してある。この要素が True に設定され、SCO が学習目標を設定しないときは、LMS は既定値により試行進捗情報を決定しなくてはならない。この要素が True に設定され、SCO が主学習目標の学習目標状態を示していないときは、LMS が既定値を用いて試行進捗情報を決定しなくてはならない。逆に、この要素が False (既定値) の場合、SCO により示される主学習目標の学習目標状態が使用される。SCO が学習目標状態を示していないなら、アクティビティは習得とみなされる。

このページは空白である。

DRAFT

6 章

シーケンシング XML バインディング

DRAFT

このページは空白である。

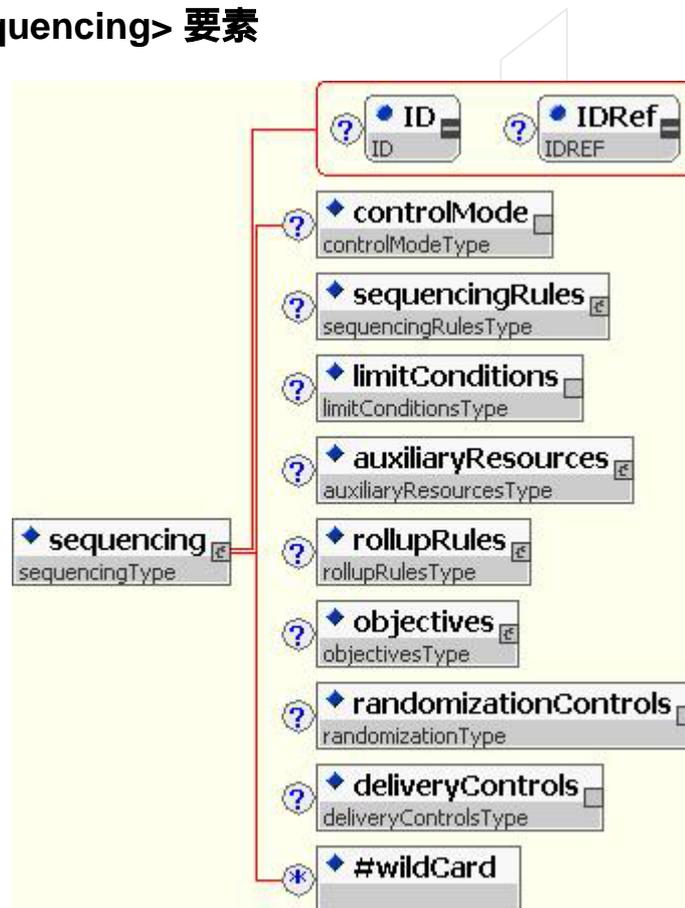
DRAFT

6.1. XML バインディングシーケンシング定義モデル

XML バインディングは IMS SS XML バインディング・バージョン 1.0 の最終版仕様と IMS SS XML スキーマ定義(XSD)に基づいている。XML バインディングは、XML へ IMS SS 定義モデルを解釈してバインドされる方法を定義する。バインディングは、定義モデルのどの項目がどの XML 要素や XML 属性等で表現されているかという事を説明する。

この章は、IMS SS XSD(imsss_v1p0.xsd)と IMS SS XML バインディング・バージョン 1.0 の最終版仕様によって定義された必要項目に加えて付属すべき SCORM の必要項目について述べる。

6.1.1. <sequencing> 要素



説明: 全てのシーケンシングルールは一番外側（ルート要素）の中に含まれる。この一番外側のルート要素は、特定のアクティビティもしくはアクティビティのセットに適用するシーケンシングルールをカプセル化する。

繰り返し: <sequencing>要素はコンテンツパッケージマニフェストの要素である<item>と<organization>の拡張として0あるいは1回のみ出現する。

属性:

- ID(0 または 1 , 任意):シーケンシングルールを一意に明確にする識別子。データ型:xs:ID
- IDRef(0 または 1 , 任意):シーケンシング要素の最新の場所を解決するのに利用される, (同じパッケージまたは submanifest 内の) <resource>識別子へのリファレンス。これによりシーケンシングルールの集合がマニフェスト全体で再利用可能となる。データ型:xs:ID

要素:

- controlMode
- sequencingRules
- limitConditions
- auxiliaryResources
- rollupRules
- objectives
- randomizationControls
- deliveryControls

列挙されたそれぞれの子要素について定義された内容モデルの詳細については, IMS SS XML バインディング文書を参考の事。

6.1.2. <sequencingCollection> 要素

一組のシーケンシングルール(ひとつの<sequencing>要素の中で定義された)が再利用される場合がある。IMS SS 仕様の XML バインディングは, シーケンシングルールの集合のコンテナの役を果たす<sequencingCollection>(シーケンシング集合)要素を定義する。<sequencing>要素の”IDRef”属性を使って, <sequencing>要素の ID を IDRef への使用に割り当てることによって再利用が起こる。この仕組みを利用するために参照される<sequencing>要素は, <sequencingCollection>要素の中に格納することができる。<sequencing>要素が, “IDRef”属性とシーケンシングルールのインライン定義の両方を使う場合, インラインで定義された任意の最上位レベルの要素は, 参照される要素内で定義された全ての同類要素に優先する。

シーケンシング集合に関して, 注意すべき重要ないくつかの特徴と必要条件がある。

1. <sequencingCollection>を使う場合, シーケンシング集合は一度だけ存在できる。
2. もし<sequencing>要素がシーケンシング集合で定義された<sequencing>要素を参照するように設定されれば, IDRef 属性は必須であり, シーケンシング集合で定義された<sequencing>要素によって指定された ID を参照するものとする。

3. もし<sequencing>要素がシーケンシング集合で定義された<sequencing>要素を参照するように設定されれば，インライン<sequencing>要素の ID はオプションである．
4. シーケンシング集合に定義されたすべての<sequencing>要素で，ID 属性は必須であり，IDRef 属性は許されていない．この必要条件はシーケンシングルール“chaining”を避けるための位置付けである．この必要条件は，将来緩められるかもしれないが，“chaining”の使用に関するもっと多くのユースケースが必要とされる．



説明: <sequencingCollection>要素は，再利用可能なシーケンシングルールの集合のコンテナの役を果たす．

繰り返し: <sequencingCollection>要素は，<imscp:manifest>要素の子として0あるいは1回存在する．

属性:

- なし

要素:

- <sequencingCollection> が使われる場合，<sequencing>要素が1回以上存在する．

6.1.3. コンテンツパッケージとの関係

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とコンテンツリソースを関係づけるための構造を提供する（*item* 要素と *resource* 要素との関係）．さらに *item* 要素は，親の *organization* 要素の中に含まれる集合としてクラスタに集める事ができ，これは学習アクティビティが親アクティビティまたは複数アクティビティと共に集まるのと同様である．従って，IMS SS は学習アクティビティの概念を，*item* 要素，*organization* 要素内の *item* 要素群，および，コンテンツパッケージング仕様によって定義される *organization* 要素自身にマップする．コンテンツパッケージング XML バインディングは，この仕様によって，シーケンシング情報をパッケージされたコンテンツに関連付ける方法を定義するために拡張される．

学習アクティビティの特定の順序を定義するプロセスは，IMS コンテンツパッケージング仕様の 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ァイルを使ってやり取りするためにコンテンツのアグリゲーションの作成で始まる．以下の図で示されるように，コンテンツパッケージング *organization* 要素とその中の各々の *item* 要素はシーケンス情報の関連を通してシーケンス動作を定義することができる．：

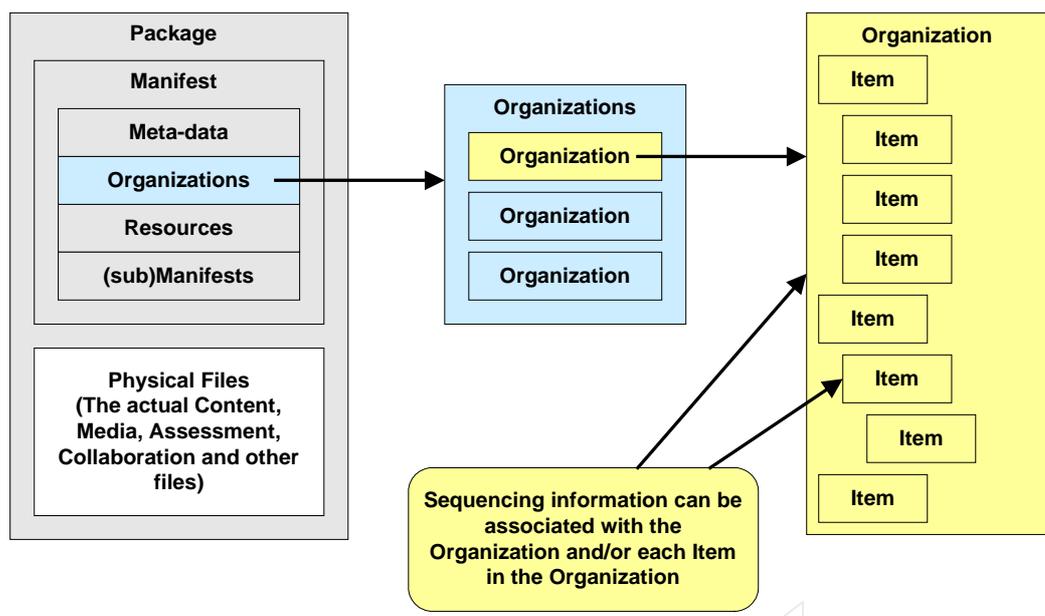


図 6. 1. 2a: コンテンツパッケージング構造

すべての SCORM 準拠コンテンツアグリゲーション・パッケージは、シーケンシング情報をデフォルトで含む。もし SCORM コンテンツ・パッケージがいかなるシーケンシングルールも含まなければ、暗示された既定の動作は、学習者にガイダンスや制約なく自由にどのアクティビティも選ばせることである。

6.1.3.1 SCORM 1.2 コンテンツパッケージプロファイルへの変更

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 コンテンツパッケージを処理するために、LMS によって必要とされるいくつかの要素を含む。それらの要素は adlcp 名前空間に含まれ、IMS コンテンツパッケージ仕様への拡張として適用される。

IMS SS 仕様は、adlcp 名前空間要素と同じ機能を提供するいくつかの要素を定義している。適切な場合、adlcp 名前空間要素はそれらの対応する IMS SS 要素で置き換えられた。

以下の一覧は、廃止要素とそれに対応する置換えについて述べる。

- **adlcp:prerequisites** . この要素は廃止され、SCORM バージョン 1.3 の manifest で使われないものとする。この要素は、IMS SS XML バインディングで定義された単独の代替要素を持たないが、その効果と動作はシーケンシングルールを組み合わせを使うことによって表現できる。
- **adlcp:masteryscore** . この要素は廃止され IMS<imsss:primaryObjective>要素上で定義される<imsss:minNormalizedMeasure>要素によって置き換えられる。(IMS SS XML バインディング 3.8.3 節参照)

-
- *adlcp:maxtimeallow* . この要素は廃止され , IMS<imsss:limitConditions>要素上で定義された属性 attemptExperiencedDurationLimit によって置き換えられる . (より詳細は IMS SS XML バインディング 3.5 節を参照) .

以下の adlcp 名前空間要素は , IMS SS 仕様を反映するために変更された .

- *adlcp:scormtype*:この要素は ADL 名前空間に残る . その機能はどの IMS SS 仕様要素にも置き換えられない . 唯一の変化は “ sca ” の固定された語彙のリストへの追加である .

DRAFT

このページは空白である。

DRAFT

7章 シーケンシング動作

DRAFT

このページは空白である。

DRAFT

7.1. シーケンシング動作

本章では、IMS SS 仕様で記述されたシーケンシングプロセスに伴う動作を説明する。以下の説明は、SS 仕様に書かれている詳細な動作の説明（擬似コード）を置換しようとしているわけではなく、それらのプロセスの主要な機能と特徴の理解を助けるものである。

IMS SS 仕様は、アクティビティツリーの個々のアクティビティに適用される二つのデータモデル、すなわち、アクティビティツリーの状態を管理するデータモデルとシーケンシング動作の集合を含んでいる。データモデルおよびそのアクティビティとの関係は、以下のように要約できる：

- **トラッキングモデル(IMS SS TM)** - アクティビティと関連づけられた学習資源と学習者とのインタラクションから集められた情報。これは、動的な実行時（学習者が LMS と対話している間）のデータモデルである。
- **アクティビティ状態モデル(IMS SS AM)** - アクティビティツリーにおける各アクティビティのシーケンシング状態と、アクティビティツリーのグローバルな状態の管理。これは、LMS の実装シーケンシングエンジンがシーケンシングセッションを通してアクティビティツリーの状態を管理するために使用する動的な実行時データモデルである。
- **シーケンシング定義モデル(IMS SS SM)** - 様々なシーケンシングプロセスが、アクティビティを「順序付け」するために、どのようにトラッキングモデル情報を利用し解釈するかの記述。これは、あるコンテンツアグリゲーションに対する作成時のシーケンシング意図を記述する静的な（コンテンツパッケージにおいて定義される）データモデルである。5 章は、シーケンシング定義モデルを詳細に説明している。

個々のアクティビティについて、トラッキングとシーケンシングの定義モデルのすべての要素の集合はアクティビティに適用され、アクティビティの状態と考えることができる。シーケンシングプロセスは、アクティビティの状態とアクティビティ状態モデル（LMS の実装シーケンシングエンジンが管理する状態）を、それらの動作を表現するために利用する。

様々なシーケンシング動作は、独立なプロセスとして記述されるが、それらは上で説明した 3 つの集合の上で動作する。個々のシーケンシング動作は、いくつかのプロセスとサブプロセスを含む。それはより詳細に動作を定義しているが、他のシーケンシング動作とは関係しない。オーバーオールシーケンシングプロセスは、すべてのシーケンシング動作が、“シーケンシングセッション”と“シーケンシンググループ”の文脈の中で、どのように互いに関連するかを定義している。

7.1.1.1. トラッキングモデル

アクティビティの条件付きシーケンシングを可能にするために、学習者とアクティビティに関連する学習資源とのインタラクションの情報は維持管理されなければならない。IMS SS 仕様は、アクティビティツリー上の各アクティビティに対して管理されるべきトラッキング情報を記述している。 - トラッキングモデルはトラッキング情報を記述している。

SCORM では、トラッキングモデルがどのように表現され、また管理されるかについて実装上の要求を課さない。さらに、トラッキングモデルが後述の要素を含む、あるいは、それに制限されるという制約はない。各アクティビティについてトラッキング情報がただ一つだけ存在しなければならないこともない。実装に対して要求しているのは唯一、IMS SS 仕様で記述される動作を、以下に述べるトラッキングモデルに作用するかのよう（as-if）に、表現することだけである。実装する際、アクティビティツリーの what-if 評価の動きに合わせて自由にトラッキング情報を管理し、使うことができる。しかしながら、さまざまなシーケンシングプロセスがオーバーオールシーケンシングプロセスの文脈で適用されるとき、プロセスは、IMS SS 仕様で記述されているように同一の妥当なトラッキング情報のセットを使わなければならない。

7.1.1.1.1 はじめに

SCORM の前のバージョンでは、定義された唯一のデータモデルは SCORM ランタイム環境データモデルであった。この情報は、SCO での学習者のインタラクションを記録するために使用された。資源と関連するアクティビティの、どんな情報がどのように記録されたかについて、いつもいくらかの混乱があった。IMS SS 仕様を加えたことによって追加されたデータモデル、トラッキングモデルは、LMS で管理するために、より明確に定義されている。トラッキングモデルは、アクティビティツリーの個々のノードと関連づけられたシーケンシング状態情報の動的な集合である。この状態データのセットは、個々の学習者ごとにアクティビティツリーの個々のノードと関連づけられる。これらのデータ項目の初期値は、SCORM コンテンツパッケージに含まれているシーケンシングルールにおいて定義される。学習実行中に、学習者がアクティビティおよび学習資源とインタラクションを行なうと、この状態情報が更新される（もし適用可能ならば）。状態情報は実装特有のものであり、定義されたシーケンシングルールとこのアプリケーションプロファイルで定義されている状態データの要求条件に基づいている。

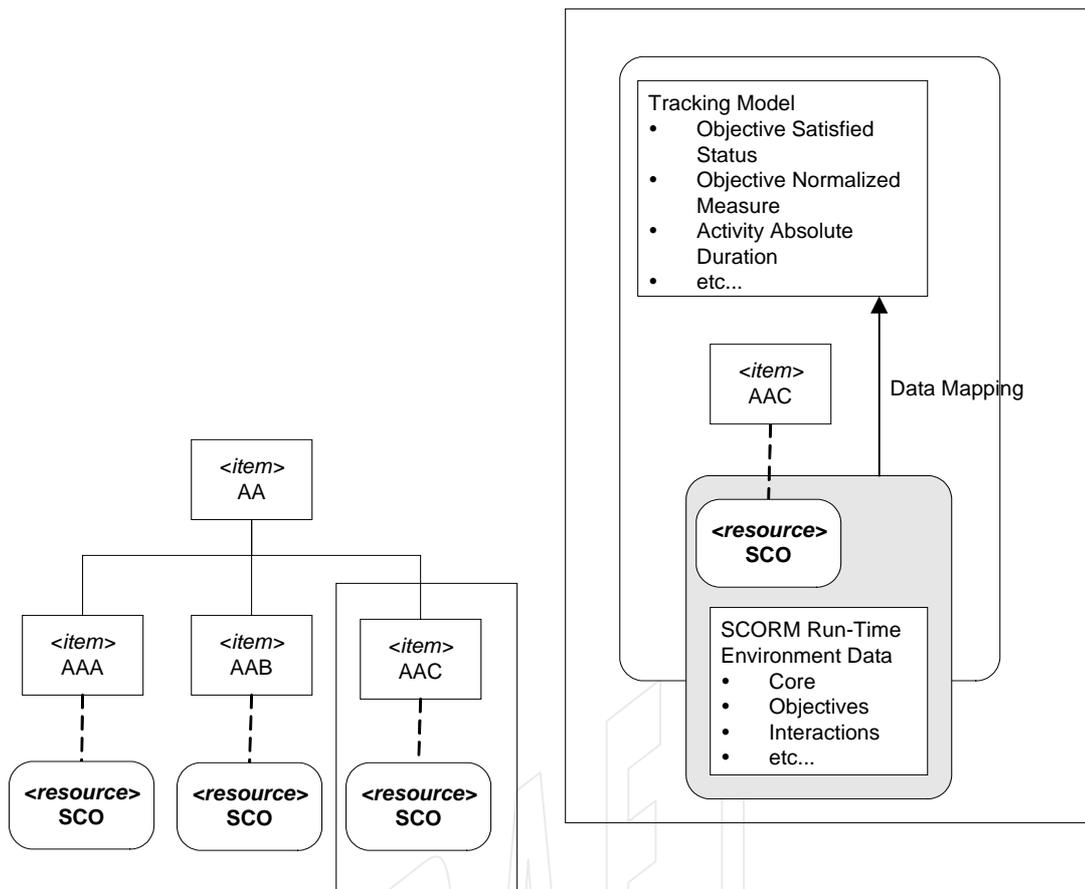


図 7.1.1.1a: ランタイム状態モデルの概念図

SCORM ver.1.2 において定義された SCORM ランタイム環境データモデルは、伝達される情報（例えば、習得状態やスコア）を定義するために学習資源（SCO）で使われる。LMS は、これらの学習資源と関連する必要なアクティビティのランタイム状態を保持しなければならない。LMS は、セッションを越えてこの状態を保持しなければならない。もし複数の LMS にまたがる再使用が起こるならば、学習資源は、これらのあらかじめ決められたデータ要素だけを利用しなければならない。SCORM ランタイム環境データモデル要素の中には、トラッキングモデルで定義される要素に大きな影響を与えるものもある。これらの関係と動作は次節で説明される。

条件付きのアクティビティシーケンシングを可能にするためには、アクティビティと関連づけられた学習資源と学習者のインタラクションについての情報を維持管理しなければならない。IMS SS 仕様は、アクティビティツリーの個々のアクティビティで保持されるべきトラッキング情報を記述している。 - トラッキング情報を記述するデータモデルの要素セットはトラッキングモデルと呼ばれる。

7.1.1.2 トラッキングモデル

IMS SS 仕様は、学習アクティビティに関する進捗と習得情報を含む習得状態の記録に基づいて、条件付きのシーケンシング動作を行なうことを可能にする。IMS SS 仕様は、実行時のシーケンシング判断の基となる習得状態情報の共通ボキャブラリを定義する。

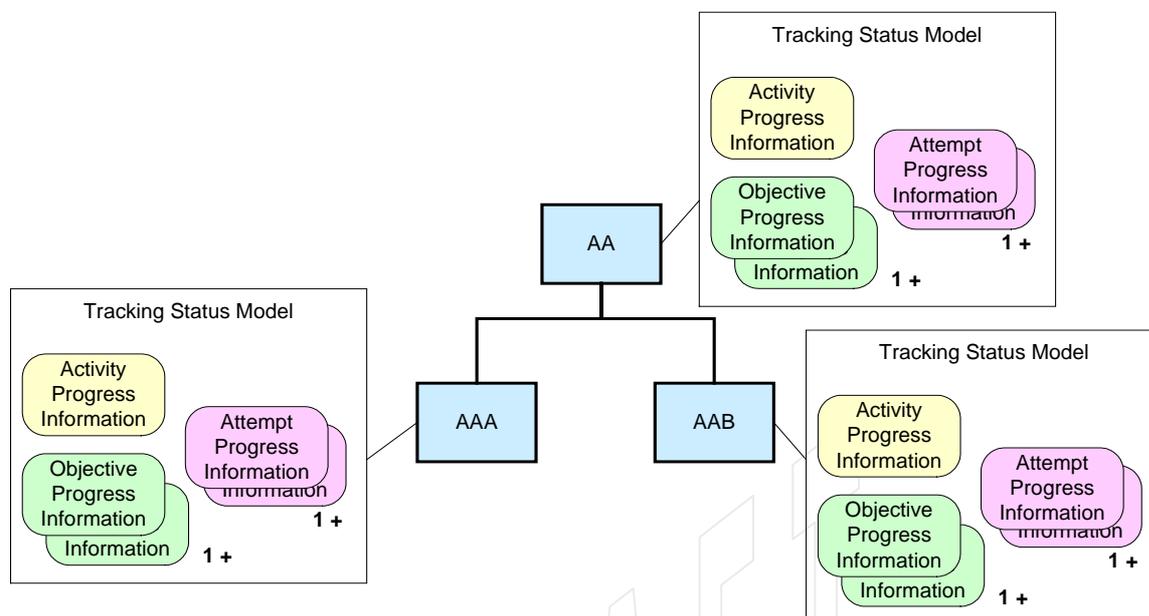


図7.1.1.2a: トラッキングモデル

アクティビティは、アクティビティを実行する個々の学習者固有の習得状態のトラッキング情報を有する。LMSは、学習者と起動された学習資源とのインタラクションの結果として、実行時にトラッキング情報を更新するものと仮定されている。通信する学習資源(SCO)と関連するアクティビティに対して、LMSは、学習資源からの通信に基づいてトラッキングモデルを管理する。この管理プロセスは、定義された学習アクティビティのすべての状態を決定し、学習アクティビティが配信されるべき順序を決定する。通信しない学習資源に対しては、次章で、LMSにおけるトラッキングモデルの管理を助けるためのいくつかの定義された必要条件をあげる。

アクティビティにおける習得状態要素の状態変化は、親アクティビティの状態に影響する。この親アクティビティの状態の変化への影響を評価するプロセスはIMS SS仕様において“ロールアップ”と呼ばれる。情報の“ロールアップ”は、LMSによって行なわれる。

コンテンツの作者によるシーケンシングルールはアクティビティへの関係付けを通して、習得状態のトラッキング情報はLMSがシーケンシング動作をコントロールするために使用される。種々のシーケンシングプロセスは作成時のシーケンシング情報とアクティビティのトラッキング状態に影響される。この依存性は、アクティビティを習得したかどうかという学習者の進捗とアクティビティ自身の状態に基づいて動的なシーケンシング動作を可能にするように定めら

れている。シーケンシングとトラッキング状態情報は、実行時にアクティビティツリーのどのアクティビティをナビゲーション要求に応じて配信するかを決定するために用いられる。

習得状態のトラッキング情報要素は、アクティビティツリーの個々のアクティビティごとに保持されなければならない。アクティビティは、より高レベルのアクティビティを形成するために他のアクティビティとともに集約される。アクティビティは、どのような深さにでも入れ子にできる。子アクティビティは、それが存在するならば、いつもその親アクティビティの文脈で実行される。

トラッキングモデルは、順序付けされたアクティビティを配信するシステムが維持しなければならない情報を記述している。LMSは、定義された個々のアクティビティについての習得状態のトラッキング情報を受け取り、維持することができなくてはならない。トラッキングモデルは、以下の習得状態のトラッキング情報のセットを定義する：

- **学習目標進捗情報(Objective Progress Information)**：学習目標に関連する学習者のインタラクションの結果。
- **アクティビティ / 試行進捗情報(Activity/Attempt Progress Information)**：アクティビティにおける学習者の進捗の記述。この情報は、個々のアクティビティに対する進捗情報の累積を記述する。

7.1.1.3 学習目標進捗情報

アクティビティは、それと関連づけられた学習目標をいくつでも持つことができる。個々の学習目標の特性は、シーケンシング定義要素 *Objective Description* によって記述される。アクティビティにおける個々の試行に対して、学習者は、そのアクティビティと関連づけられた個々の学習目標についての学習目標進捗情報（表 7.1.1.3a）の一組を取得する。

No.	名称	説明	値空間	既定値
1	<i>Objective Progress Status</i>	学習目標が習得状態値を有することを示す（True または False）。	論理型	False
2	<i>Objective Satisfied Status</i>	学習目標が習得されていることを示す（True または False）。習得されているか、いないかの決定や意味は、このモデルにおいて定義されない。 <i>Objective Progress Status</i> が True でない限り、値は意味を持たない。	論理型	False
3	<i>Objective Measure Status</i>	学習目標が習得度を有することを示す（True または False）。	論理型	False
4	<i>Objective Normalized Measure</i>	-1 から 1 の間で正規化された学習目標の習得度（例えばスコア）。習得値を正規化するメカニズムは、このモデルにおいて定義されない。 <i>Objective Measure Status</i> が True でない限り、値は意味を持たない。	-1 から 1 の実数 有効数字 4 桁以上。	0.0

表7.1.1.3a: 学習目標進捗情報

学習目標進捗情報の表は各要素に対し既定値を定めている。既定値は、LMSの実装シーケンシングエンジンによって陽に設定されるまで使用される。

学習目標進捗情報は、ペアのデータモデルを利用する。ひとつの要素は、記録された情報を記述し、もうひとつは、その記録された情報が有効であるかどうかを記述する。例えば、*Objective Satisfied Status* は、学習目標が満たされているかどうかを記述し、*Objective Progress Status* は、*Objective Satisfied Status* の値が有効であるかどうかを記述する。詳細なシーケンシング動作は、ペアのデータモデルの両方の値を参照する。トラッキング情報はランタイムデータモデルであるので、実装者はシステムを最適化するために、これらの値を自由に表現することができる；しかし、それらのシステムは、IMS SS 擬似コードで記述された動作を示さなければならない。

この仕様書をより理解しやすくするために、個々のペアを記述するためにただ1つの要素を用い、‘不定’値をボキャブラリに追加する。例えば *Objective Satisfied Status* は、以下のボキャブラリを使って記述される：

- **satisfied** – *Objective Progress Status* は True , *Objective Satisfied Status* は True に設定される。
- **not satisfied** – *Objective Progress Status* は True , *Objective Satisfied Status* は False に設定される。
- **unknown** – *Objective Progress Status* は False に設定される。

同様に、その通常の浮動小数点範囲に加えて、*Objective Normalized Measure* は、*Objective Measure Status* が False に設定されることを表す‘不定’値を持つとして記述される。

IMS SS 仕様は、ローカル目標とグローバル共有目標を区別している。特に指定がない場合、学習目標ごとに、アクティビティにおける試行ごとに初期化される学習目標進捗情報の集合は、そのアクティビティに対して“ローカル”である。しかしながら、シーケンシング定義要素 *Objective Map* は、“ローカル”な学習目標をグローバル共有目標とリンクさせている。以下にあげたルールは、シーケンシングプロセスが学習目標の状態にアクセスするときどの情報が使われるかを記述している。このルールには、順序がある；最初に適用されるルールに関連する情報が使われ、次のルールは無視される。

1. *Objective Satisfied Status* が要求され、ローカル目標が *Objective Satisfied Status* を定義する（ローカル目標に対して *Objective Progress Status* が True）ならば、ローカルな *Objective Satisfied Status* が使用される。
2. *Objective Satisfied Status* が要求され、ローカル目標が *Objective Satisfied Status* を定義せず（ローカル目標に対して *Objective Progress Status* が False）、*Read Objective Satisfied Status* の学習目標マップがグローバル共有目標にローカル目標をリンクすると定義されるならば、グローバル共有の *Objective Satisfied Status* が使用される（その有効性は考慮しない）。

3. *Objective Normalized Measure* が要求され、ローカル目標が *Objective Normalized Measure* を定義する（ローカル目標に対して *Objective Measure Status* が True）ならば、ローカルな *Objective Normalized Measure* が使用される。
4. *Objective Normalized Measure* が要求され、ローカル目標が *Objective Normalized Measure* を定義せず（ローカル目標に対して *Objective Measure Status* が False）、*Read Objective Normalized Measure* の学習目標マップがグローバル共有目標にローカル目標をリンクすると定義されるならば、グローバル共有な *Objective Normalized Measure* が使用される（その有効性は考慮しない）。

注意：シーケンシングプロセスにおいて、状態情報がグローバル共有目標から取り出されるとき、ローカル目標の状態は変更できない。

アクティビティにおける試行が終了するとき、“write” 学習目標マップが適用される。もし、Write Objective Satisfied Status と（あるいは）Write Objective Normalized Measure が True である（既定値は False）と特定されたとき、適切な学習目標進捗情報（*Objective Satisfied Status* と *Objective Normalized Measure*）が、ローカル目標から関連するグローバル共有目標にコピーされる。グローバル共有目標がいかなる状態であっても、無条件に上書きされる。

7.1.1.4 アクティビティ進捗情報

それぞれのアクティビティは、そのアクティビティにおけるすべての試行にわたる一組の情報を持っている；この情報がアクティビティ進捗情報である（表 7.1.1.4a）。

No.	名称	説明	値空間	既定値
1	<i>Activity Progress Status</i>	アクティビティについてのアクティビティ進捗情報が有意義であること（True または False）を示す。	論理型	False
2	<i>Activity Absolute Duration</i>	アクティビティにおける試行の累積時間、すなわちアクティビティの最初の開始から終わりまでの時間。経過時間を決定するためのメカニズムはこのモデルにおいて定義されない。 <i>Activity Progress Status</i> が True でない限り、値は意味を持たない。	期間 0.1 秒きざみ	0.0
3	<i>Activity Experienced Duration</i>	アクティビティにおけるすべての試行の <i>学習経過時間</i> の累積、すなわちアクティビティが中断されている時間を含まないアクティビティの開始から終わりまでの時間（すなわち、アクティビティが実行されていないか、または不活性な時）。中断時間の経過時間を決定するためのメカニズムは、このモデルにおいて定義されない。 <i>Activity Progress Status</i> が True であり、アクティ	期間 0.1 秒きざみ	0.0

		ビティが葉でない限り，値は意味を持たない．		
4	<i>Activity Attempt Count</i>	アクティビティにおける試行の回数．カウントは現在の試行を含む．すなわち，0は，アクティビティが試行されていないことを意味し，1以上は，進行中または完了を意味する． <i>Activity Progress Status</i> が True でない限り，値は意味を持たない．	負の値でない整数	0

表 7.1.1.4a: アクティビティ進捗情報

アクティビティ進捗情報の表は，各要素の既定値を定義する．既定値は，LMSの実装シーケンシングエンジンによって陽に設定されるまで使用される．

アクティビティ進捗データモデル要素は次のように管理される：

- *Activity Progress Status* は，アクティビティについての最初の試行が始まる時に，True に設定される．
- *Activity Absolute Duration* は，アクティビティにおけるすべての試行の絶対経過時間の総和である．
- *Activity Experienced Duration* は，アクティビティにおけるすべての試行の学習経過時間の総和である．この値は葉でないアクティビティについては記録されない．
- *Activity Attempt Count* は，そのアクティビティにおけるそれぞれの新しい試行が始まる時にインクリメントされる．

7.1.1.5 試行進捗情報

アクティビティの個々の試行に対して，学習者は一組の試行進捗情報を取得する（表 7.1.1.5a）．

No.	名称	説明	値空間	既定値
1	<i>Attempt Progress Status</i>	アクティビティの試行に対して，試行進捗情報（True または False）が有意味であることを示す． <i>Activity Attempt Count</i> が 0 より大きくない限り，値は信頼できない．値は意味を持たない．	論理型	False
2	<i>Attempt Completion Amount</i>	0 から 1 の間で正規化された，アクティビティにおける試行の完了の評価値．1 は，アクティビティの試行が完了していることを意味し，それ未満は，完了していないことを意味する．完了の程度を定義するメカニズムは，このモデルにおいて定義されない． <i>Attempt Progress Status</i> が True でない限り，値は意味を持たない．	1 から 1 の実数．有効数字 4 桁以上．	0.0
3	<i>Attempt Completion Status</i>	アクティビティの試行が完了していることを示す（True または False）．完了か未完了かの決定や意味は，このモデルにおいて定義されない．	論理型	False

		<i>Attempt Progress Status</i> が True でない限り，値は意味を持たない．		
4	<i>Attempt Absolute Duration</i>	アクティビティにおける試行の経過時間．すなわち試行の開始から終わりまでの時間．経過時間を決定するためのメカニズムはこのモデルにおいて定義されない． <i>Attempt Progress Status</i> が True でない限り，値は意味を持たない．	期間 0.1 秒きざみ	0.0
5	<i>Attempt Experienced Duration</i>	アクティビティの学習経過時間，すなわちアクティビティ試行が中断されている間の経過時間を含まない，試行の開始から終わりまでの時間．（すなわち，アクティビティ試行が実行されていないか，または不活性な時）．経過時間または中断時間を決定するためのメカニズムは，このモデルにおいて定義されない． <i>Attempt Progress Status</i> が True でない限り，値は意味を持たない．	期間 0.1 秒きざみ	0.0

表 7.1.1.5a: - 試行進捗情報

試行進捗情報の表は，各要素に対して既定値を定義している．既定値は，LMS の実装シーケンシングエンジンによって陽に設定されるまで使用される．

アクティビティ進捗データモデル要素は，以下のように管理される．

- *Attempt Progress Status* は，他のいくつかのトラッキング情報がアクティビティにおける現在の試行に対して記録されるとき，True に設定される．
- *Attempt Completion Status* は，アクティビティにおける現在の試行が完了しているかどうかを記述する．
- *Attempt Completion Amount* は，アクティビティにおける現在の試行の完了の程度を記述する．IMS SS 仕様の現在のバージョンは，*Attempt Completion Amount* を使用しない．SCORM は，この要素に対するいかなる追加動作も定義しない．実装は，自由にこの要素を使用することができる．
- *Attempt Absolute Duration* は，アクティビティにおける現在の試行の絶対経過時間の総和である．
- *Attempt Experienced Duration* は，アクティビティにおける現在の試行の学習経過時間の総和である．

試行進捗情報は，ペアのデータモデルを使用する．ひとつの要素は，記録された情報を記述し，もうひとつは，記録された情報が有効であるかどうかを記述する．例えば，*Attempt Completion Status* は試行が完了したかどうかを記述し，*Attempt Progress Status* は *Attempt Completion Status* の値が有効であるかどうかを記述する．詳細なシーケンシング動作は，ペアのデータモデルの両方の値を参照する．トラッキング情報は，ランタイムデータモデルなので，実装者はシス

テムを最適化するために、これらの値を自由に表現することができる；しかし、それらのシステムは、IMS SS 擬似コードで記述された動作を表現しなければならない。

この文書をより理解しやすくするために、個々のペアを記述するためにただ1つの要素を用い、「不定」値をボキャブラリに追加する。例えば、*Attempt Completion Status* は、以下のボキャブラリを使って説明される：

- **completed** –*Attempt Progress Status* は True , *Attempt Completion Status* は True に設定される。
- **incomplete** –*Attempt Progress Status* は True , *Attempt Completion Status* は False に設定される。
- **unknown** –*Attempt Progress Status* は False に設定される。

同様に、定義された経過時間のタイプに加えて、*Attempt Absolute Duration* と *Attempt Experienced Duration* は、*Attempt Progress Status* が False であることを表すために、「不定」値を持つように記述される。

7.1.1.6 アクティビティ状態情報

シーケンシングエンジンは、アクティビティツリー上の個々のアクティビティについての追加の状態情報を維持する。この情報は、アクティビティ状態情報と呼ばれる（表 7.1.1.6a）；それは、オーバーオールシーケンシングプロセスにおける様々なシーケンシングプロセスによって利用され、管理される。

No.	名称	説明	値空間	既定値
1	<i>Activity is Active</i>	アクティビティにおける試行が進行中であることを示す。つまり、アクティビティが学習者に配信されて、終了されていないか、または <i>Current Activity</i> の祖先である（True または False）。	論理型	False
2	<i>Activity is Suspended</i>	現在アクティビティが中断されていることを示す（True または False）。	論理型	False
3	<i>Available Children</i>	アクティビティにおいて使用可能な子アクティビティの順番を示しているリスト。	アクティビティの順序つきリスト	All children

表 7.1.1.6a: アクティビティ状態情報

アクティビティ状態データモデル要素は次の通り管理される：

- *Activity is Active* は、アクティビティにおける試行が始まる時に True に設定され、試行が終わる時に False に設定される。この要素は、様々なシーケンシングプロセスの文脈において、いくつかの特徴と影響をもつ：
 - 1回の試行では、アクティビティツリーには1つの“アクティブパス”だけが存在する。アクティブパスに沿ったアクティビティだけが、*Activity is Active* を True に設定することができる。“アクテ

「イブパス」は、ツリー構造のルートから始まり、*Current Activity* で終わる。(次章のグローバル状態情報を見よ)

- ただ1つの葉アクティビティだけが、*Activity is Active* を True に設定することができる(または存在しない)。もし葉アクティビティが、*Activity is Active* が True ならば、そのアクティビティは *Current Activity* であるはずである。
- *Current Activity* が終了していない(その現在の試行が終了していない)ならば、*Current Activity* だけが、*Activity is Active* を True に設定できる。
- *Activity is Suspended* は、アクティビティにおける現在の試行が終わる時に、True に設定される。これは、アクティビティのタイプによる2つの方法のうちのどちらかに従って行なわれる。
 - もしアクティビティが葉であるならば、アクティビティに関連づけられた学習資源や LMS は、中断によってアクティビティの学習資源が終了されたことを示す。
 - アクティビティが親クラスタである場合、LMS の実装シーケンシングエンジンは、その子のどれかが中断されているとき、クラスタに中断を設定する。

アクティビティの中断状態は、どのようにアクティビティにおける次の試行を初期化するかを記述する。もし、アクティビティが中断されていれば、アクティビティにおける次の試行は、前の試行を再開し、前のトラッキングモデル状態を使用する。このとき新しいトラッキング情報は初期化されない。

- *Available Children* は、LMS の実装シーケンシングエンジンにおけるシーケンシング動作の実行で利用可能なアクティビティの子を配列したリストを保持する。この要素は親のクラスタを表現するアクティビティにだけ適用される。この要素は、ナビゲーション動作、終了動作、ロールアップ動作、シーケンシング動作、配信動作の間に、順序付けされるべき子のセットとして、暗に利用される。

注意：LMS の実装シーケンシングエンジンは、クラスタについて作成時に定義されたアクティビティの配列リストを保持しなければならない。このリストは、選択とランダム動作において *Available Children* 属性に影響するために使用される。

7.1.1.7 グローバル状態情報

LMS の実装シーケンシングエンジンは、アクティビティツリーに対する追加の状態情報を維持する。この情報は、グローバル状態情報と呼ばれる(表 7.1.1.7a)；それは、オーバーオールシーケンシングプロセスを通して様々なシーケンシングプロセスによって利用され、管理される。

No.	名称	説明	値空間	既定値
1	<i>Current Activity</i>	現在のアクティビティを示す。 もしアクティビティが学習者により実行中ならば、現在のアクティビティは、最後に完了したコンテンツ配信環境プロセスにより提供されたアクティビティである（配信動作を見よ）。 もしアクティビティが学習者により実行中でないならば、現在のアクティビティは、最後に完了された終了プロセスによって終了と識別されたアクティビティである（終了動作を見よ）	Activity	None
2	<i>Suspended Activity</i>	Suspend All ナビゲーション要求を生じたアクティビティを示す。	Activity	None

表7.1.1.7a: グローバル状態情報

- *Current Activity* は、LMS の実装シーケンシングエンジンによって現在記録されているアクティビティツリー上の唯一のアクティビティを示す。このアクティビティの祖先はすべて、*Activity is Active* が True でなければならない。*Current Activity* は、すべてのシーケンシング要求が実行される場所を定義する。

注意：シーケンシング要求が IMS SS 仕様で *Current Activity* から処理されるとして定義されている一方、実装においては唯一の要素を使用することは要求されない。実装では、IMS SS 仕様で記述されている動作に従いさえすればよい。例えば、ある実装では、もっとも最近配信されたアクティビティを *Current Activity* とし、最も最近終了したアクティビティを *First Candidate Activity*（シーケンシング要求プロセスの開始アクティビティ）として記録したいかもしれない。

- *Suspended Activity* は、前のシーケンシングセッション(SuspendAll ナビゲーション要求で終了したシーケンシングセッション)で *Current Activity* であった唯一のアクティビティを示す。すべてのアクティビティのアクティビティツリーのルートから *Suspended Activity* へのパスは、*Activity is Suspended*（上記アクティビティ状態を見よ）がすべて True に設定される。続くシーケンシングセッションは、パスに沿ったすべてのアクティビティ（*Suspended Activity* を含む - これは、「ブックマーキング」のひとつの形式とみなせる）を再開する *Resume All* ナビゲーション要求で開始する。

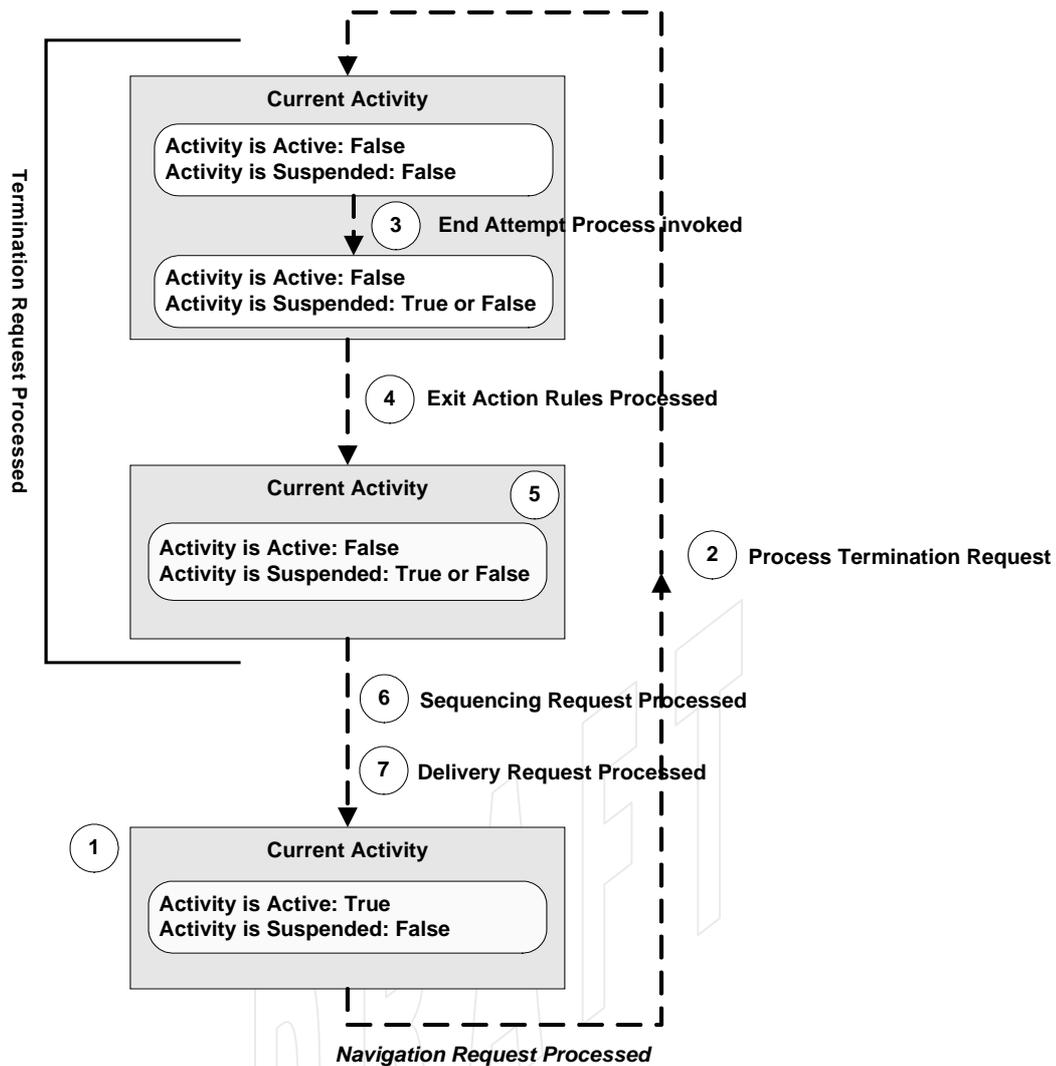


図7.1.1.7a: *Current Activity* 状態モデル

【図中】

- ナビゲーション要求の処理
- 処理の終了要求
- 試行プロセス完了の呼び出し
- 終了アクションルールの処理

- シーケンシング要求の処理
- 配信要求の処理

図 7.1.1.7a は、*Current Activity* の状態モデルを表す；それは、様々なシーケンシングプロセスの *Current Activity* および現在のアクティビティの状態に与える影響を示している。

1. LMS の実装シーケンシングエンジンは、学習資源が学習者に配信されたことを仮定する。資源に関連するアクティビティは *Current Activity* である。 - それはアクティビティツリーの葉アクティビティであるはずであり、今、アクティブである。 *Current Activity* の（「アクティブパス」に沿った）祖先すべてもアクティブであり、アクティビティにおける試行は、アクティビティの祖先における試行の文脈の中で起こる。
2. アクティビティツリーの状態は、あるナビゲーション要求がオーバーオールシーケンシングプロセスを引き起こすまでこのように変わらない。もしナビゲーション要求が有効ならば、それは *Current Activity* での試行を終わるための終了要求を結果として生じる。 - そのアクティビティ状態属性 *Activity is Active* は False となる。 *Current Activity* は、葉アクティビティのままである。
3. 葉アクティビティが終了する時、システムは、終了状態がアクティビティの中断であることを示すかもしれない。 - これは、試行終了プロセスで行なわれる。システムがこれを示すとき、葉アクティビティの *Activity is Suspended* の値は True となる。
4. 終了動作プロセスでは、葉アクティビティの祖先すべてにおける終了動作ルールが評価される。これはシーケンシング終了動作ルールサブプロセスで行なわれる。このサブプロセスの結果、葉アクティビティが *Current Activity* であり続けるか、または、葉アクティビティの祖先が *Current Activity* になる。
5. 終了動作プロセスでは、ポスト条件ルールが *Current Activity* において評価される。もし、*Current Activity* が中断されていなければ、それは葉アクティビティか終了アクションルールサブプロセス（後述）で決定されるアクティビティである。
6. シーケンシング動作は、未解決のシーケンシング要求を処理する。それは結果として配信要求を生じることがある。
7. 配信動作プロセスは、未解決の配信要求を有効にする。もし配信要求が有効ならば、*Content Delivery Environment Process* が呼び出される。このプロセスにおいて、ターゲットアクティビティが *Current Activity* になり、そこで試行が開始する。 - *Current Activity* のアクティビティ状態属性 *Activity is Active* は True になり、*Activity is Suspended* は False となる。
8. ステップ 1 に戻る。

7.1.1.8 トラッキング動作

この章の情報は、IMS SS 仕様のトラッキングモデル動作の章を置き換えるわけではなく、補うことを意図している。詳細については IMS SS 仕様を参照せよ。

特に指定がない場合、個々の学習目標に対する学習目標情報のセットだけが、その学習目標を定義したアクティビティに適用される。これらは“ローカルな”学習目標と呼ばれる。他のアクティビティは、別のアクティビティに関連づけられた学習目標の情報を直接参照することはできない。複数のアクティビ

ティが同じ学習目標の情報を参照することを可能にするためには、両方のアクティビティについて *Objective Map* が定義されなければならない。個々の *Objective Map* は、“ローカルな”学習目標と、あるグローバル共有目標との関係を定義している。マップには2種類が存在する。：

- “write” *Objective Map* は、対応するグローバル共有目標のフィールドを、“local”目標の値に設定する。この操作は、アクティビティにおける試行が終了するときに行われる。

注意：複数のローカル目標は、同じグローバル共有目標に状態を「書き込む(write)」ことができない。この動作は、未決定動作を引き起こす。

- “local”目標の値が LMS の実装シーケンシングエンジンから要求されるが未定な時 (*Objective Measure Status* が False , または *Objective Progress Status* が False) にはいつでも、“read” *Objective Map* は、対応するグローバル共有目標のフィールドを読む。この操作では、“local”目標情報を変更しない。

注意：ローカル目標は、ひとつのグローバル共有目標から「読む(read)」ことだけができる。でなければ、未決定の動作が引き起こされる。

7.1.2. オーバーオールシーケンシングプロセス

この節の情報は IMS SS 仕様の補足情報である。詳細についてはそちらを参照していただきたい。

オーバーオールシーケンシングプロセスはシーケンシングエンジンに包括的な制御プロセスを示している。これはシーケンシングセッションの文脈の中で様々なシーケンシング動作がどのように適用されるかを記述したものである。オーバーオールシーケンシングプロセスでは、以下のシーケンシング動作をまとめている。

- ナビゲーション動作：ナビゲーション要求をどのように確認し、終了要求とシーケンシング要求に解釈するかを記述している。
- 終了動作：アクティビティ上の現在の試行をどのように終了するか、アクティビティツリーの状態をどのように更新するか、現在の試行の終了時にどんなアクションを行うべきかを示している。
- ロールアップ動作：クラスタのトラッキング情報をどのようにしてクラスタの子ノードのトラッキング情報から導くかを示している。
- 選択ランダム化動作：シーケンシング要求を処理している際に、クラスタ内のアクティビティをどのように考慮するかを示している。
- シーケンシング動作：配信される「次」のアクティビティを特定するために、試行内のアクティビティツリー上でシーケンシング要求をどのようにして処理するかを示している。

- 配信動作：配信のために特定されたアクティビティの配信のための正当性を確認し，LMS がそのアクティビティをどのように扱うべきかを示している．

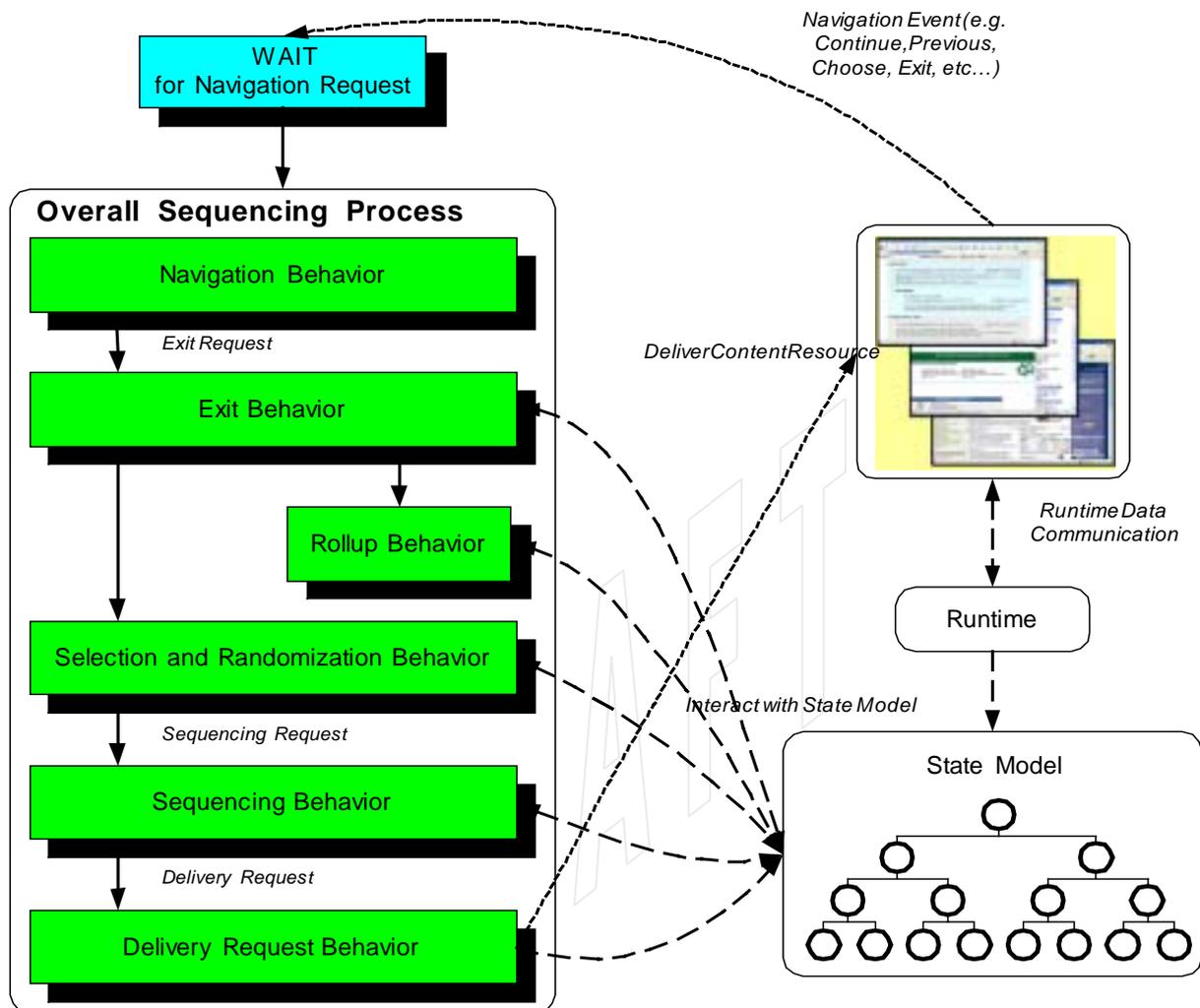


図7.1.2a - Conceptual Model of the Overall Sequencing Process

IMS SS仕様では，以上の動作を同じモデル - トラッキングモデル（7.1.1.2節参照）に作用する独立で単体のプロセスとして記述している．図7.1.2aに示すオーバーオールシーケンシングプロセスの概念モデルは，様々なシーケンシングアクティビティ間の関係やアクティビティツリーとトラッキングモデルの関係を描いている．オーバーオールシーケンシングプロセスの開始点はLMSから発せられるナビゲーション要求である．一般に，ナビゲーション要求は学習者が引き起こしたナビゲーション制御に関して生成される（7.13節 ナビゲーション動作参照）．オーバーオールシーケンシングプロセスの終了点は「次に」配信するアクティビティの決定か，ある中間状態である．オーバーオールシーケンシング動作の結果は学習者に配信されるものか，中間状態の表示である．

オーバーオールシーケンシングプロセスはシーケンシングセッションの開始と終了の時にシーケンシンググループの動作を見せる。シーケンシンググループの各ステップを以下に詳細に記述する。

7.1.2.1 シーケンシンググループ

シーケンシングセッションの開始。

- (1) 学習者がコンテンツ配信環境や LMS にアクセスし（例，システムにアクセスする，ログインする），ある教授ユニットでコンテキストを確立する（例，コースやコンテンツアグリゲーションを選ぶ）。
- (2) LMS は，もしそのコンテンツが初めて実行される場合には *Start* ナビゲーション要求，*Resume All* ナビゲーション要求，*Choice* ナビゲーション要求を発することによってシーケンシングプロセスを開始する。
- (3) ナビゲーション動作プロセスは *Start* ナビゲーション要求，*Resume All* ナビゲーション要求，*Choice* ナビゲーション要求を適切なシーケンシング要求に変換し，処理する。

シーケンシンググループの開始

- (4) トラッキングモデル内やシーケンシング要求の情報を使って，シーケンシング動作は学習者に配信する適切なアクティビティ（ツリーの中のノード）を定めるために，アクティビティツリーをトラバースする。
- (5) 配信動作は選択されたアクティビティが配信できるかを決定し，配信できる場合は，そのアクティビティのコンテンツリソースを学習者に配信する準備をする。配信できない場合は，オーバーオールシーケンシングプロセスは停止し，他のナビゲーション要求があるまで待つ。
- (6) 学習者は学習資源を利用する。その間，シーケンシングプロセスは何もせず，要求を待つ。
- (7) 学習者が学習資源を利用する間，アクティビティはトラッキングモデルの様々な要素を更新する値を報告する。
- (8) 学習者，配信システム，アクティビティは *Continue*, *Previous*, *Choose activity X*, *Abandon*, *Exit* といったナビゲーションイベントを起こす。
- (9) LMS はナビゲーション要求を発することによってシーケンシングプロセスにナビゲーションイベントを知らせる。
- (10) ナビゲーション動作はナビゲーション要求を終了要求やシーケンシング要求に変換する。ナビゲーション要求が学習者がセッションを終了したいということを示していたら，シーケンシングセッションは完了する（シーケンシングセッションの終了動作やアクティビティ状態モデルの保持方法は指定されておらず，実装にゆだねられている）。
- (11) アクティビティが終端や終了によってナビゲーション要求を引き起こすと，トラッキングモデルを更新する付加的な値を報告する。その後，アクティビティは終了する。ロールアッププロセスは，アクティビティのコンテンツリソースの配信結果として起こる状態変化の影響を決

定するために起動される。そのプロセスはそのアクティビティやアクティビティツリー内の親アクティビティのためにトラッキングモデルを更新する。

- (12) ステップ4のはじめからプロセスを繰り返す。

オーバーオールシーケンシングプロセスは様々なシーケンシングプロセスがどのように関係しているかを示しているが、実装においては、オーバーオールシーケンシングプロセスのコンテキストの外側で、個々のシーケンシングプロセスを自由に呼び出すことができる。ただし、その場合には、アクティビティやアクティビティツリーのトラッキングモデルが、ナビゲーションイベントが発生したときにオーバーオールシーケンシングプロセスで記述された動作を示すことを保証するように十分な状態管理機構を提供しなければならない。オーバーオールシーケンシングプロセスの外側からシーケンシングプロセスを呼び出す一般的なシナリオは、有効な学習リソースを開始するナビゲーション要求だけが生じることを保証するか、妥当なナビゲーション制御だけを許す「知的」ユーザインタフェースを提供するために、ナビゲーション要求の有効性を確認することである。

実装において、シーケンシング動作がオーバーオールシーケンシングプロセスの外側から「もし~だったら」というシナリオを評価するために、「不正なデータ」もしくは（複数のトラッキング情報を管理することによる）仮のデータで呼び出されてもかまわない。例えば、ある実装が学習者に継続ナビゲーションイベントを起こすことを許すナビゲーション制御を提供しているかもしれない。学習者がその制御を起こすと、LMSはその仮の継続ナビゲーション要求を処理して何が起こるかを見せる。もし仮のデータがエラーを起こしたり、何も配信しなければ、LMSはそのナビゲーションイベントを無視し（オーバーオールシーケンシングプロセスを呼ばない）、学習者に通知した上で、その“不正な”データを破棄する。仮の要求が成功した場合には、LMSはオーバーオールシーケンシングプロセスを呼び出し（もしくはその一部を最大限に活用し）学習リソースを配信し、仮の状態データを確定する。

7.1.3. ナビゲーション動作

ナビゲーション動作はオーバーオールシーケンシングプロセスの主たる入り口である。これは学習者やシステムにシーケンシングエンジンと通信する手段を提供する。ナビゲーション意図を示す外部イベントはナビゲーションイベントとよばれる。これらのイベントを引き起こす手段はナビゲーション制御とよばれる。LMSはナビゲーションイベントを処理し、シーケンシングエンジンに対して対応するナビゲーション要求を発する責任がある。

7.1.3.1 ナビゲーションイベント

ナビゲーションイベントは（LMSのシーケンシングエンジンの実装の）外側のイベントで、いくつかの方法でコンテンツをナビゲートする学習者やシステムの意図を表す。これらのイベントは一般に学習者によるユーザインタフェース制御を通じて引き起こされるが、LMSがナビゲーションイベントを引き起こすこともできる。SCORMはどのようにナビゲーションイベントが引き起こされるかについては制約しない。

ナビゲーションイベントが検出されると、LMSは次の二つのうちのいずれかで応答する。

1. イベントを無視する LMSは、ナビゲーションイベントが学習者にとって望ましくないシステム状態（実行）を引き起す場合には、そのイベントを無視する。SCORMはナビゲーションイベントが何も配信しないときにLMSがどのように振る舞うかについて要求しない。例えば、現在、学習者がアクティビティツリーの最後の葉に関する学習リソースを実行（経験、利用）しているならば、次の項目を続けてほしいという要求に対して、なにも配信されない。LMSは“continue”ナビゲーションイベントを無視する。
2. ナビゲーション要求を発する - LMSはナビゲーションイベントを有効なナビゲーション要求に変換し、オーバーオールシーケンシングプロセスを呼び出す。

7.1.3.2 ナビゲーション制御

ナビゲーション制御は学習者に特定の方法で *Current Activity* から離れることを要求する手段を提供するユーザインタフェースである。SCORMはLMSやコンテンツに対して、どのナビゲーション制御を利用可能にするか、どのように提示されるか、どのように引き起こされるか、どんなナビゲーションイベントを引き起こすかは制約しない。

SCORM準拠なLMSは最低限、対応するナビゲーションイベントが有効なナビゲーション要求に変換されるようなナビゲーション制御を提供しなければならない。例えば、（現在分かっているアクティビティツリーの状態から）ユーザの選択に対して有効な対象となるアクティビティがあるとき、LMSは学習者がそのアクティビティを選択する手段をいくつか提供しなければならない。

SCORM準拠のLMSとコンテンツはナビゲーション制御について対応するイベントが結果として有効ではないナビゲーション要求に変換されるか、なにも配信されない場合はナビゲーション制御を提示しなくてもよい。例えば、*Current Activity* がアクティビティツリーのルートであるとき、LMSはpreviousナビゲーション要求を引き起こす手段を提供しなくともよい。LMSがコンテンツが対応するイベントが結果として不正なナビゲーション要求を導いたり、配信するコンテンツが無いようなナビゲーション制御を提示するとき、LMSは適切な中間ページを答えなければならない。

7.1.3.3 ナビゲーション要求

オーバーオールシーケンシングプロセスは、ナビゲーション要求がナビゲーションエンジンに発せられたときに開始する。一度、ナビゲーション要求が発せられると、IMS SS 仕様プロセスで記述されたすべての動作が適用されなければならない。

SCORM LMS は以下のナビゲーション要求を受け入れ、対応する動作を示さなければならない。(表 7.1.3.3a).

ナビゲーション要求	動作
<i>Start</i> (開始)	<i>Current Activity</i> が未定義の場合、 <i>Start</i> シーケンシング要求を発する。
<i>Resume All</i> (再開)	<i>Current Activity</i> が未定義で、the <i>Suspended Activity</i> (A.M.1.2)が定義されている場合、 <i>Resume All</i> シーケンシング要求を発する。
<i>Continue</i> (次画面)	<i>Current Activity</i> の <i>Activity is Active</i> の値が True であれば、 <i>Exit</i> 終了要求を発する。 <i>Continue</i> シーケンシング要求を発する。
<i>Previous</i> (前画面)	<i>Current Activity</i> の <i>Activity is Active</i> (A.M.1.2)の値が True であれば、 <i>Exit</i> 終了要求を発する。 <i>Previous</i> シーケンシング要求を発する。
<i>Forward</i>	このバージョンの SCORM では明示されない
<i>Backward</i>	このバージョンの SCORM では明示されない
<i>Choice</i> (選択)	<i>Current Activity</i> の <i>Activity is Active</i> の値が True であれば、 <i>Exit</i> 終了要求を発する。 <i>Choice</i> シーケンシング要求を発する。その要求はターゲットアクティビティの指定を伴う。
<i>Exit</i> (終了)	<i>Exit</i> 終了要求を発する。 <i>Exit</i> シーケンシング要求を発する。 <i>Current Activity</i> の現在の試行が正常に終了される。アクティビティの終了は他の外部ナビゲーションイベント(例えば、 <i>Continue</i> , <i>Previous</i> , <i>Choice</i> など)の結果ではない。
<i>Exit All</i> (全て終了)	<i>ExitAll</i> 終了要求を発する。 <i>Exit</i> シーケンシング要求を発する。
<i>Suspend All</i> (中断)	<i>Suspend All</i> 終了要求を発する。 <i>Exit</i> シーケンシング要求を発する。 <i>Current Activity</i> の現在の試行とそのすべての親が正常に終了し、その試行は完了せず、そのアクティビティは習得にならない。そのアクティビティは将来再開される(再開は新しい試行にならない)。LMS のシーケンシングエンジンは、将来そのアクティビティが再開されるかもしれないので、必要な状態とトラッキング情報を記録しなければならない。
<i>Abandon</i> (中止)	<i>Abandon</i> 終了要求を発する。 <i>Exit</i> シーケンシング要求を発する。 <i>Current Activity</i> の現在の試行が異常に終了し、そのアクティビティは習得にならない。そのアクティビティを再開することができない。トラッキングデータによるロールバックはない。

Abandon All (全て中止)	Abandon All 終了要求を発する。 Exit シーケンシング要求を発する。 配信されているアクティビティの現在の試行とそのすべての親が異常に終了し、そのアクティビティは習得にならない。そのアクティビティを再開することができない。トラッキングデータによるロールバックはない。
--------------------	--

表7.1.3.3a: SCORM 1.3 ナビゲーション要求

7.1.3.4 ナビゲーション動作プロセス

この節の情報は IMS SS 仕様のナビゲーション動作の節の補足情報であり、置き換えるものではない。詳細についてはそちらを参照していただきたい。

ナビゲーション動作プロセスはオーバーオールシーケンシングプロセスによって引き起こされるが、LMS によって直接呼び出されることもある。動作プロセスはナビゲーション要求を受け入れ、有効ではないナビゲーション要求、シーケンシング要求、もしくは終了要求とシーケンシング要求を返す。

有効ではないナビゲーション要求が返される時を以下にまとめる:

- 定義されていない (表 7.1.3.3a にない) 要求を受け取ったとき。
- *Start* ナビゲーション要求を受け取ったが、シーケンシングセッションが既に始まっている
- *Resume All* ナビゲーション要求を受け取ったが、シーケンシングセッションが既に始まっている。もしくは、シーケンシングセッションは始まっていないが、*Suspended Activity* (前のシーケンシングセッションが *Suspend All* ナビゲーション要求で終了していない) が存在しないとき。
- *Continue* ナビゲーション要求を受け取ったが、*Current Activity* の親の *Sequencing Mode Flow* の値が *true* という状態になっていない。
- *Previous* ナビゲーション要求を受け取ったが、*Current Activity* の親の *Sequencing Mode Flow* の値が *True* に設定されていない、もしくは *Control Mode Forward Only* の値が *False* に設定されていない。
- *Choice* ナビゲーション要求を受け取ったが:
 - その対象がアクティビティツリー内に存在しない。これは目的アクティビティがツリー内に全く存在しないか、対象の親の選択可能な子の中に対象が入っていない場合である (選択ランダム化動作を参照のこと)。
 - 対象の親の *Sequencing Control Mode Choice* の値が *False* に設定されている。
 - *Choice* ナビゲーション要求を処理することが (アクティブなパス上で) アクティブなアクティビティを終了させることを要求し、そのアクティビティの *Sequencing Control Mode Exit* の値が *False* という状態になっている。

有効な *Continue*, *Previous* ないし *Choice* ナビゲーション要求は対応するシーケンシング要求に変換される。加えて、有効な *Continue*, *Previous* ないし *Choice* ナビ

ゲーシヨン要求が発せられ、*Current Activity* がアクティブであれば、*Current Activity* の現在の試行を終了するために *Exit* (終了)ナビゲーシヨン要求が発せられる。

Exit, *Exit All*, *Suspend*, *Abandon*, *Abandon All* ナビゲーシヨン要求はすべて、対応する終了要求と *Exit* シーケンシヨン要求となる。これらのナビゲーシヨン要求は、*Current Activity* の試行を終了し、場合によってはシーケンシヨンセッションを終了させる。

7.1.4. 終了動作

終了動作には二つの意図がある。すなわち、*Current Activity* 現在の試行を終了し、アクティビティツリーの状態について最新で正確な状態を確保する。終了動作は終了要求に基づいて行われる。終了動作は *Current Activity* を動かし、シーケンシヨン要求を返す。

アクティビティの終了とそのアクティビティに関連する学習資源の終了を区別することが重要である。アクティビティに関連する学習資源がいつどのようにして終了するかは SCORM の範囲外である。SCORM は SCO が終了する前に通信を終了する (*LMSFinish()*とよぶ) ことだけを要求する。アクティビティは内部シーケンシヨン動作の一部として終了する。これは学習資源の終了に影響されないが、(LMS の実装によって) 関連する学習資源を終了することを要求する場合がある。

より具体的に言うと、*Current Activity* がアクティブな場合、それは終了要求に応じて終了する。アクティビティツリーがシーケンシヨン要求を処理する前に最新で正確な状態であるように、シーケンシヨンエンジンの実装が *Current Activity* が終了していることを保証しなければならない。LMS の実装が、もっとも最新の正確な情報がシーケンシヨンエンジンで利用可能か確認するために、終了アクティビティに関連する学習資源も終了することを要求する場合もある。

7.1.4.1 終了要求

一般的に、終了要求は *Current Activity* の試行が終了しなければいけない、つまり、*Current Activity* が不活性になる、ということを示す。IMS SS 仕様はいくつかの種類の終了要求を定義しており、それぞれ違う動作を導く。SCORM 準拠の LMS は以下の動作を実行する。(表 7.1.4.1a).

終了要求	動作
Exit	<i>Current Activity</i> の現在の試行が正常に終了させられる。
Exit All	アクティブなアクティビティ (ルートから <i>Current Activity</i> までのもの) の現在の試行は正常に終了させられる。
Suspend All	アクティブなアクティビティ (ルートから <i>Current Activity</i> までのもの) の現在の試行は中断させられる。

Abandon	<i>Current Activity</i> の現在の試行は異常に終了させられ，そのアクティビティは完了にならない．そのアクティビティを再開することができない．トラッキングデータのロールバックはない．
Abandon All	アクティブなアクティビティ（ルートから <i>Current Activity</i> までのもの）の現在の試行は異常に終了させられ，そのアクティビティは完了にならない．そのアクティビティを再開することができない．トラッキングデータのロールバックはない

表7.1.4.1a: SCORM 1.3 Exit Requests

7.1.4.2 事後条件 (Post Condition) と終了動作 (Exit Action) ルールの評価

アクティビティは一つ以上の事後条件 (post condition) シーケンシングルールと終了動作 (exit action) シーケンシングルールを持つ．シーケンシングルールは次のような構造を持つ．

If condition set Then action

condition set は条件の集合を定義し，これはそのアクティビティで記録している習得状況情報によって評価される．*condition set* が True であれば *action* が実行される．

シーケンシングルールアクションは，一般的に評価のタイミングによって3つの集合に分けられる．終了動作シーケンシングルールは終了動作のシーケンシング終了アクションルールサブプロセスの間のみ評価される．事後条件シーケンシングルールは終了動作の事後条件シーケンシングルールサブプロセスの間のみ評価される．

ルールの例

- **If not satisfied Then retry** – アクティビティの学習目標の状態が非習得であれば，そのアクティビティで *Retry* シーケンシング要求を処理する．
- **If outside available time range Then exit** – 現在の時間がそのアクティビティを利用できる時間を超しているならば，そのアクティビティで終了アクションシーケンシングプロセス要求を処理している間にアクティビティを終了する．
- **If attempted Then exit parent** – アクティビティが試みられていたら，このアクティビティの親を終了する．
- **If attempted Then exit all** – アクティビティの目標状態が習得であれば，アクティビティツリーを終了し，現在のシーケンシングセッションを終了する．

上記の例は定義されるシーケンシングルールの種類のほんの一部である．

7.1.4.3 終了要求プロセス

この節の情報は IMS SS 仕様の終了動作の節の補足情報であり，置き換えるものではない．詳細についてはそちらを参照すること

終了要求プロセスは *Current Activity* 試行を終了するために、オーバーオールシーケンシングプロセスによってシーケンシング要求を処理する前に呼び出される。*Current Activity* の現在の試行は以下の3つの方法のどれかで終了する。

- 正常終了 - これは *Exit* 終了要求または *Exit All* 終了要求によって起こる。*Current Activity* に関連する学習リソースがそのアクティビティのトラッキング情報に影響を与える。終了要求が *Exit All* ならば、そのシーケンシングセッションが終了する。
- 異常終了 - これは *Abandon* 終了要求または *Abandon All* 終了要求によって起こる。カ *Current Activity* に関連する学習活動はそのアクティビティのトラッキング情報に影響を与えない。終了要求が *Abandon All* ならば、そのシーケンシングセッションが終了する。
- 中断 - これは *Suspend All* 終了要求によって起こる。*Current Activity* の試行とその親のすべてが中断され、シーケンシングセッションが終了する。将来、*Resume All* ナビゲーション要求によって中断された試行が再開されるかもしれないことを意味している。その際、学習者は中断した *Current Activity* を経験することによってシーケンシングセッションを開始する。

シーケンシングセッションの間でもっとも一般的な終了要求は *Exit* である。終了要求プロセスは *Exit* 終了要求の間、以下のようなアクションを実行する。

終了試行プロセスの間

- (1) *Current Activity* の現在の試行が終了する。(*Current Activity* に対する *Activity is Active* の値が *False* に設定される)
- (2) そのアクティビティに関連する学習リソースがそのアクティビティのトラッキング情報に影響を与えるステータス情報を報告する。
- (3) そのアクティビティに関連する学習リソースがステータス情報を報告しない場合には、LMS のシーケンシングエンジンの実装がそのアクティビティのトラッキング情報を設定する。
- (4) *Current Activity* の現在の試行が正常に終了した場合には、その試行は「中断」状態で終了していたかもしれない(*Activity is Suspended* の値が *True* に設定される)。そのアクティビティに関連する学習リソースがこの状態を報告する。
- (5) ロールアップが実行される。 - *Current Activity* からのトラッキング情報が *Current Activity* の祖先を通じてアクティビティツリーに伝播される。

シーケンシング終了アクションルールサブプロセスの間

- (6) *Current Activity* の先祖の一つを終了させ、*Current Activity* にすることで、終了アクションルールがそのアクティビティ(カレントアクティビティの先祖の一つ)で定義される。

事後条件ルールサブプロセスの間

-
- (7) *Current Activity* が中断されない場合には、その事後条件ルールが評価される。これらのルールは *Current Activity* の先祖を終了させたり (*Exit Parent* や *Exit All* ルール)、シーケンシング要求を示す (*Continue, Previous, and Retry* ルール)。*Current Activity* の先祖の一つが終了すると、それが *Current Activity* になり、事後条件ルールが評価される (これは再帰的な処理である)。シーケンシング要求が示されると、その要求がオーバーオールシーケンシングプロセスに返され、未決定のシーケンシング要求が書き換えられる。

IMS SS 仕様では、終了動作を *Current Activity* の開始点についての情報を保持せずに *Current Activity* に移動する「絶対的な」処理として記述している。この動作はシーケンシング動作 (7.1.7 節参照) が *Current Activity* からすべての処理を始めるので要求される。様々なシーケンシングプロセスでは *Current Activity* の現在の試行がすでに終了し、アクティビティツリーの状態が更新されていること想定している。さらに配信動作 (7.1.8 節参照) は *Current Activity* の現在の試行が、学習リソースを開始するための未決定の配信要求を処理し、それに関連するアクティビティの試行を始める前に終了していることを想定している。

Current Activity についての情報を保持し、*Current Activity* の「仮の」終了を実行するために終了動作を拡張することは自由である。追加のトラッキングモデルや (サブ) プロセス、拡張・代替サブプロセスの利用は自由である。SCORM 準拠の LMS で唯一求められていることは、オーバーオールシーケンシングプロセスが呼び出されたときに IMS SS 仕様で記述されているように動作することである。すなわち、ナビゲーション要求が正当でオーバーオールシーケンシングプロセスが呼び出されると LMS が決定すると、保留されているシーケンシング要求を処理する前に、終了動作に記述されているように *Current Activity* の現在の試行が終了するよう見えるということである。

7.1.4.4 試行終了プロセス

試行終了プロセスはアクティビティが正常に終了するときに呼び出されるユーティリティプロセスである。このプロセスは終了アクティビティの状態が更新されることを保証する。試行終了プロセスは *Current Activity* を変更しない。

以下のアクションが試行終了プロセスで起こる:

- 終了アクティビティがアクティブではなくなる (*Activity is Active* の値が False に設定される)
- 終了アクティビティが葉の場合には、そのアクティビティに関連する学習リソースが、そのアクティビティにおいてもっとも最近の試行が「中断」状態で終了したことを示す場合がある。
- アクティビティがクラスタで、子の一つでも「中断」状態であれば、そのアクティビティも「中断」状態になる
- 終了するアクティビティが葉で、それに関連する学習リソースが SCO であれば、定義されている SCORM データマッピングによって試行終了プ

ロセス擬似コード（詳細はIMS SS 仕様を参照のこと）のステップ 1.2 以降がすぐに行われる。

- 終了するアクティビティが葉で，学習資源が習得状態や完了状態値を設定しない場合，LMS シーケンシングエンジンの実装はアクティビティのロールアップ目標を習得，完了状況を完了に設定する．これはアクティビティの *Delivery Control* の値に基づいて LMS シーケンシングエンジンの実装が自動的に行う。

7.1.5. ロールアップ動作

アクティビティツリーの各アクティビティは，そのアクティビティに対する学習者の応答を記録することができる．各アクティビティに付随するトラッキング情報はトラッキングモデル(7.1.1 節)で定義されている．葉アクティビティは学習者が直接経験する学習資源を提供することができる．学習資源は状態情報をやり取りすることができ，この情報は対応するアクティビティのトラッキング情報を設定するために使用される．学習資源が状態情報をやり取りしない時（*Objective Set by Content* が false か *Completion Set by Content* が false で示される），LMS のシーケンシングエンジンは対応するアクティビティのトラッキング情報を直接設定する．しかし，クラスタアクティビティは学習資源を提供できず，状態情報を直接設定することもない．クラスタアクティビティの状態は，その子の状態に基づいており，クラスタの状態情報を決定するプロセスは‘ロールアップ’と呼ばれる．図 7.1.5a は 3 つの子を持つクラスタを示す．この図は様々なロールアッププロセスを示すために本節を通して使用される．クラスタアクティビティ（AA）の状態情報は，子（AAA, AAB, AAC）の状態情報から，ロールアップによって決定される．

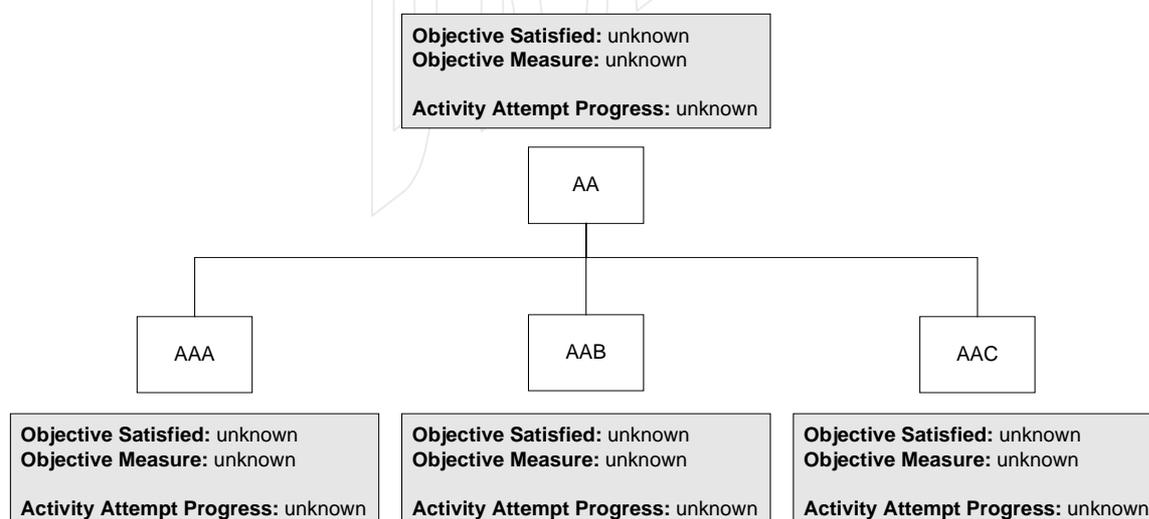


図7.1.5a: ロールアップにおいて用いられるアクティビティ状態情報

本節を通して ‘ ロールアップ ’ という用語は以下の意味で用いられる: “ クラスタアクティビティの状態情報をその子の状態に基づいて決定するプロセス ” . これは “ オーバオールロールアッププロセスを適用する ” のと同義である .

7.1.5.1 オーバオールロールアッププロセス

この節の情報は IMS SS 仕様のロールアップ動作の置換えでなく補足を意図している . 詳細については IMS SS 仕様を参照されたい .

オーバオールロールアッププロセスは , ロールアップがアクティビティツリーにどのように適用されるかを記述している . これは , すべてのロールアップ動作が適切に適用されることを保証するための制御プロセスである . オーバオールロールアッププロセスは , LMS のシーケンシングエンジンによって終了動作プロセスの間に起動されるが , LMS はそれをいつ起動しても良い . もし , LMS がオーバオールロールアッププロセスを終了動作プロセス以外で起動した場合 , LMS はアクティビティツリー中のすべてのアクティビティ (および関連する共有グローバル学習目標) のトラッキング状態が , 定義されたシーケンシング動作と整合していることを保証しなくてはならない . すなわち , ロールアップによって算出されたトラッキング状態情報は , ロールアップがオーバオールシーケンシングプロセス中で起動された場合にのみ確定されなくてはならない . LMS が ‘ 仮の ’ ロールアップを起動しても , アクティビティツリーのトラッキング状態に不都合な影響を及ぼさないために , 第二のトラッキング状態情報の組 , ‘ ダーティ ’ フラグ , ロールバック , その他の方法を実装することが考えられる .

アクティビティに関連する他のシーケンシング情報と他のシーケンシング動作のコンテキスト内で :

- ロールアップの対象は , 記録を取る (*tracked*) 子だけである .
- ロールアップの対象は , *Rollup Controls* によりロールアップに関与すると定義された子だけである .
- ロールアップは , 記録を取りかつ関与するすべての子が , 確定した状態情報を有している場合にのみ実行される .
- ロールアップは , (状態の変化により) ロールアップを引き起こした葉アクティビティの親から始まり , アクティビティツリーのルートに向かって実行される .
- 常に習得度ロールアップが最初に実行され , 次に学習目標ロールアップおよびアクティビティ進捗ロールアップが任意の順番で実行される .
- 習得度ロールアップおよび学習目標ロールアップでは , それぞれの子アクティビティ固有の学習目標でロールアップに関与するもののトラッキング情報だけが対象となる .
- 学習目標ロールアップでは , クラスタ固有の学習目標でロールアップに関与するものだけが更新される .

- オーバオールロールアッププロセスは，クラスタアクティビティの状態が変化しなくなれば終了できる．
- ロールアップルールは，クラスタアクティビティに対して，どのようにロールアップを実行するかを定義する．
- 葉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は効力を持たない．すなわち，葉アクティビティにはロールアップするものが無い．
- ロールアップはクラスタアクティビティのトラッキング情報にのみ影響する．ロールアップによってシーケンシングルールが起動されたり，その他の副作用を引き起こすことは無い．

7.1.5.2 ロールアップルールの評価

クラスタアクティビティはひとつないし複数のロールアップルールを有する．ロールアップルールは以下のような構造となっている．

If *child-activity set*, *condition set* **Then** *action*

child-activity set は記録を取られかつロールアップに関与する子アクティビティの数である．*condition set* は条件の集合で，このルールは *child-activity set* 中の各子アクティビティのトラッキング状態情報に対して評価される．もし，*condition set* が True なら，ロールアップによってクラスタのトラッキング情報は *action* に従って設定される．

以下に例を示す．

- **If any not satisfied Then not satisfied** – もし記録を取られかつロールアップに関与する子アクティビティで学習目標状態が非習得のものが一つであれば，クラスタの学習目標状態は非習得に設定される．
- **If 3 satisfied Then satisfied** – もし記録を取られかつロールアップに関与する子アクティビティで学習目標状態が習得のものが3つあれば，クラスタの学習目標状態は習得に設定される．
- **If all satisfied or completed Then completed** – もし記録を取られかつロールアップに関与するすべての子アクティビティの学習目標状態が習得ないしアクティビティ進捗状態が完了であれば，クラスタのアクティビティ進捗状態は完了に設定される．
- **If all satisfied and attempted Then satisfied** – もし記録を取られかつロールアップに関与するすべての子アクティビティで学習目標状態が習得でかつ試行されていれば，クラスタの学習目標状態は習得に設定される．
- **If 50% attempted Then incomplete** – もし記録を取られかつロールアップに関与する子アクティビティの50%でアクティビティ進捗状態が未完了であれば，クラスタのアクティビティ進捗状態は未完了に設定される．

上の例は，定義可能なロールアップルールのごく一部を示している．

7.1.5.3 習得度ロールアッププロセス

習得度ロールアッププロセスは，クラスタの習得度をクラスタの子の習得度の重み付き平均に設定する．*tracked* および *Rollup Objective Satisfied* が True の子のみが対象となる．習得度ロールアッププロセスは，クラスタの学習目標や進捗状態には影響しない．もし，もし記録を取られかつロールアップに關与する子のいずれかの習得度が未確定ならば，クラスタの習得度は未確定である．*Rollup Objective Measure Weight* を 0.0 に設定することにより，そのアクティビティの習得度をロールアップの対象から除くことができる．図 7.1.5.3a は習得度ロールアップの例である．破線の箱の情報は対応するアクティビティのシーケンシング情報を示している．

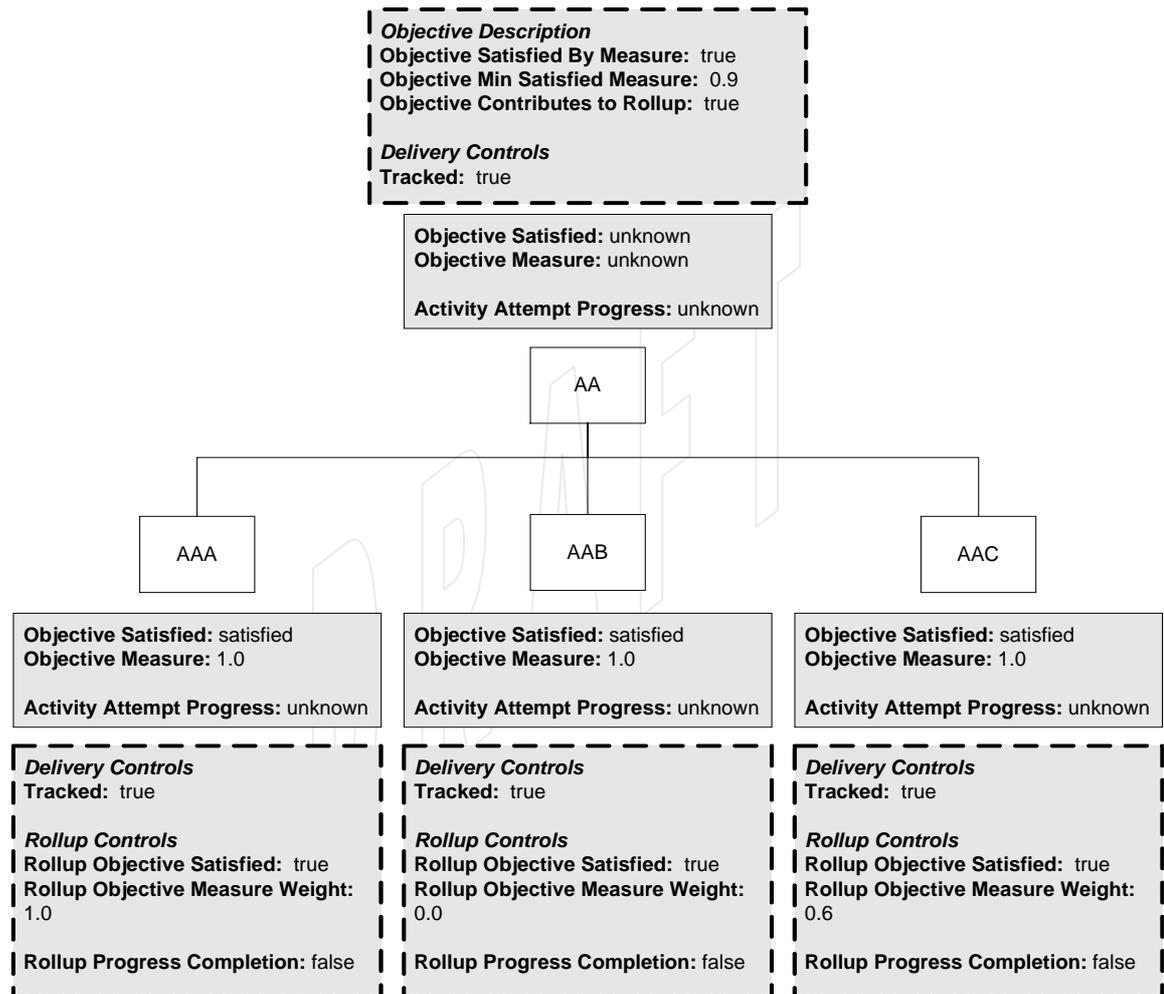


図 7.1.5.3a: 習得度ロールアッププロセスの例

7.1.5.4 学習目標ロールアッププロセス

学習目標ロールアッププロセスでは，クラスタのロールアップ学習目標（*Objective Contributes to Rollup* が true の学習目標）の状態を，未確定，習得，非習得のいずれかに設定する．*tracked* および *Rollup Objective Satisfied* が True の子のみが対象となる．学習目標情報をロールアップするために三つの方法があ

る。最初に適用された方法だけが、クラスタの学習目標状態を決定するために用いられる。

1. 習得度の利用 – もしロールアップ学習目標が *Objective Satisfied by Measure* を有していれば、ロールアップされた習得度を *Objective Minimum Satisfied Measure* と比較する。
 - もしロールアップされた習得度が未確定なら、学習目標の状態は未確定である。
 - もしロールアップされた習得度が *Objective Minimum Satisfied Measure* より大きいか等しいなら、学習目標の状態は習得となる。
 - もしロールアップされた習得度が *Objective Minimum Satisfied Measure* より小さいなら、学習目標の状態は非習得となる。

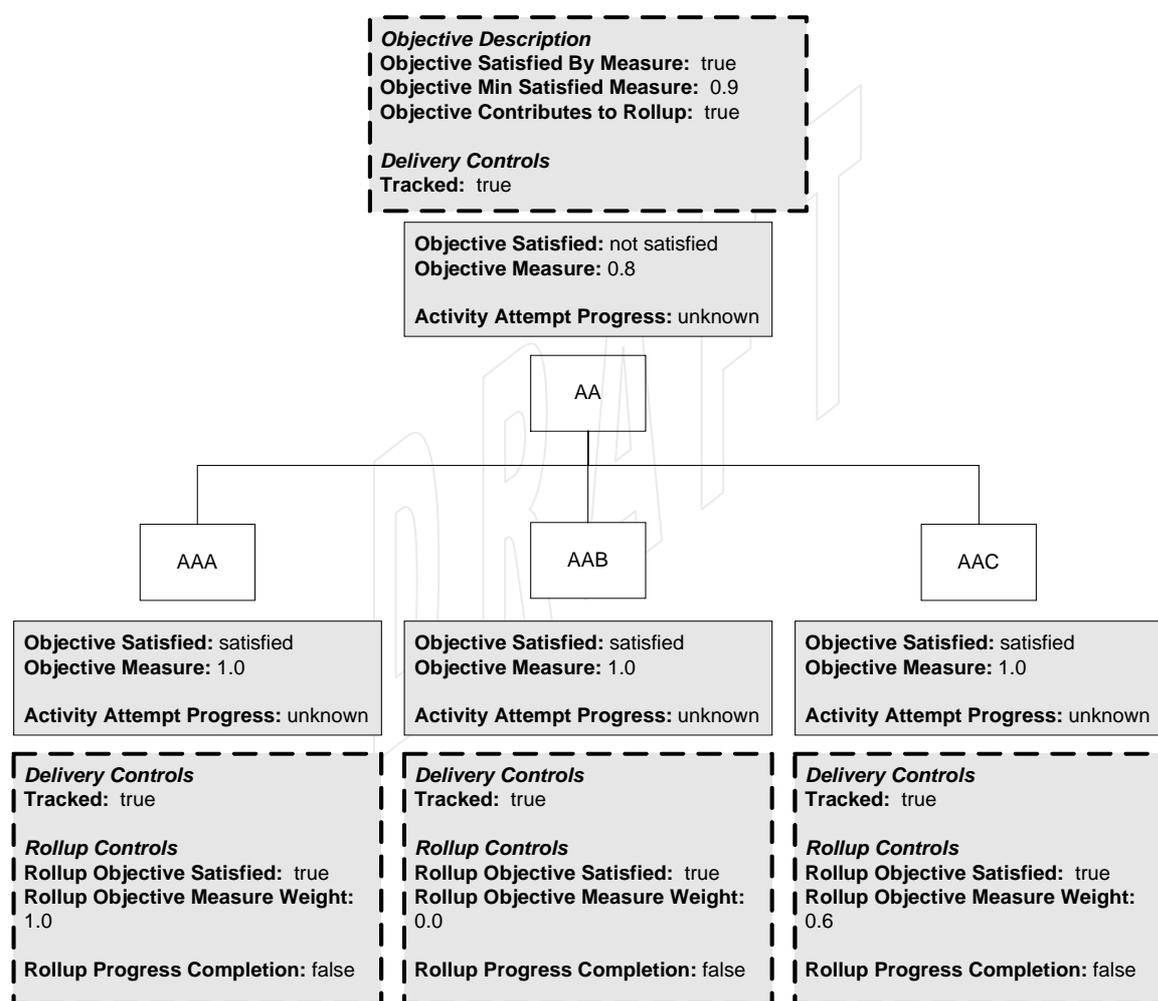


図7.1.5.4a: 習得度を使った学習目標ロールアップ

2. ルールの利用 – もしアクティビティにアクションが *satisfied* か *not satisfied* のルールが定義されていれば、クラスタの学習目標状態を決定するため

にそれらのルールが評価される。アクションが *Not satisfied* のルールが最初に評価される。

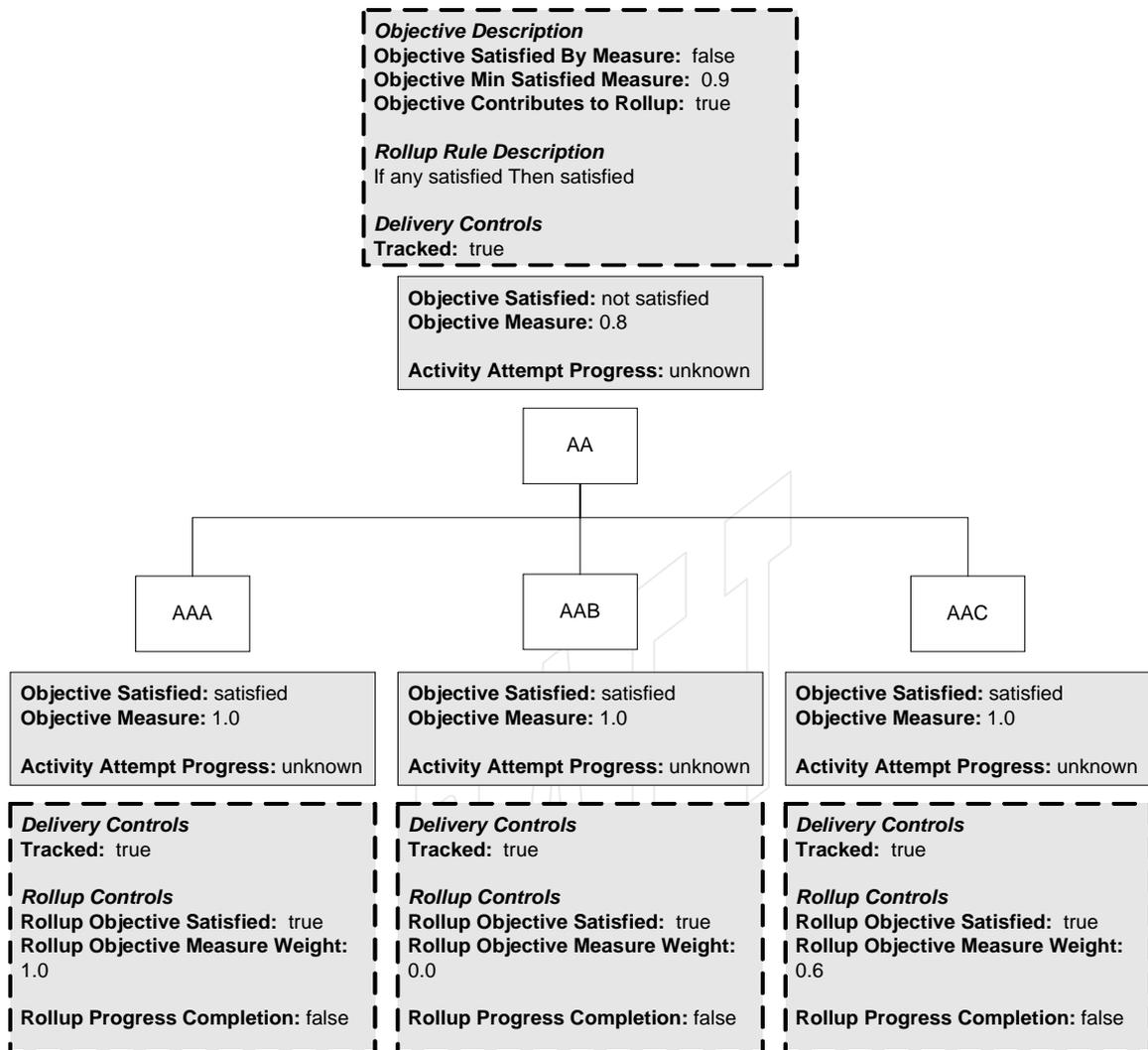


図7.1.5.4b: ルールを使った学習目標のロールアップ

3. 既定ルール – もしアクティビティにアクションが *satisfied* か *not satisfied* のルールが定義されていなければ、以下の既定のロールアップルールが評価される。
 - If all satisfied, Then satisfied: すべての子アクティビティが習得なら習得
 - If all (attempted or not satisfied), Then not satisfied: すべての子アクティビティが実行されるか非習得なら非習得

既定ルールはロールアップルールが評価されるのと同じように評価される – つまりアクションが *not satisfied* のルールが最初に評価される。

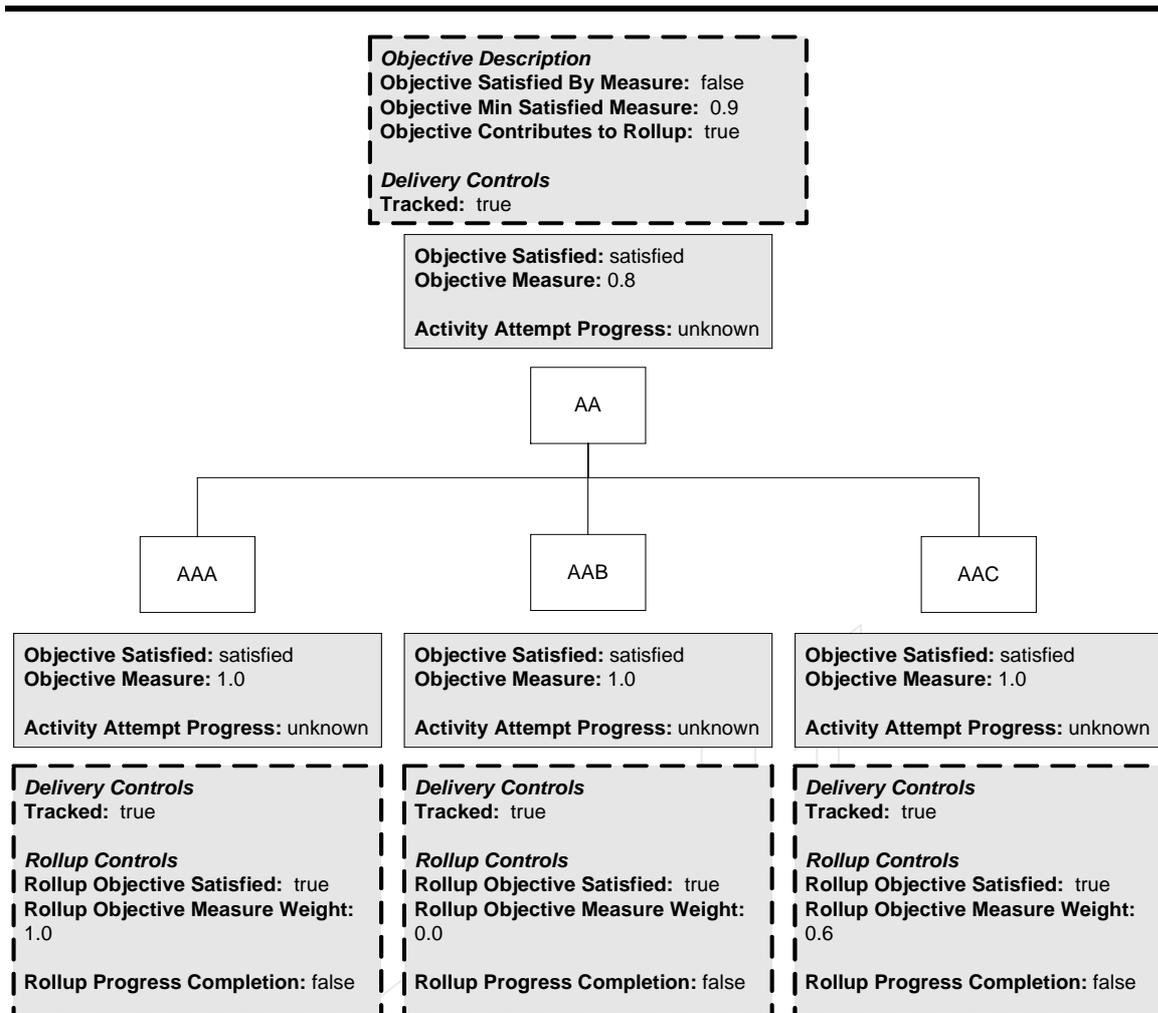


図7.1.5.4c: 既定ルールを使った学習目標ロールアップ

7.1.5.5 アクティビティ進捗ロールアップ

アクティビティ進捗ロールアップは、クラスタのアクティビティ試行進捗状態を、未確定、完了、非完了のいずれかに設定する。trackedおよび Rollup Progress Completion が True の子のみが対象となる。進捗状態をロールアップするために二つの方法がある。最初に適用された方法だけが、クラスタの進捗状態を決定するために用いられる。

1. ルールの利用 – もしアクティビティにアクションが complete か incomplete のルールが定義されていれば、クラスタの進捗状態を決定するためにこれらのルールが評価される。アクションが Incomplete のルールが最初に評価される。

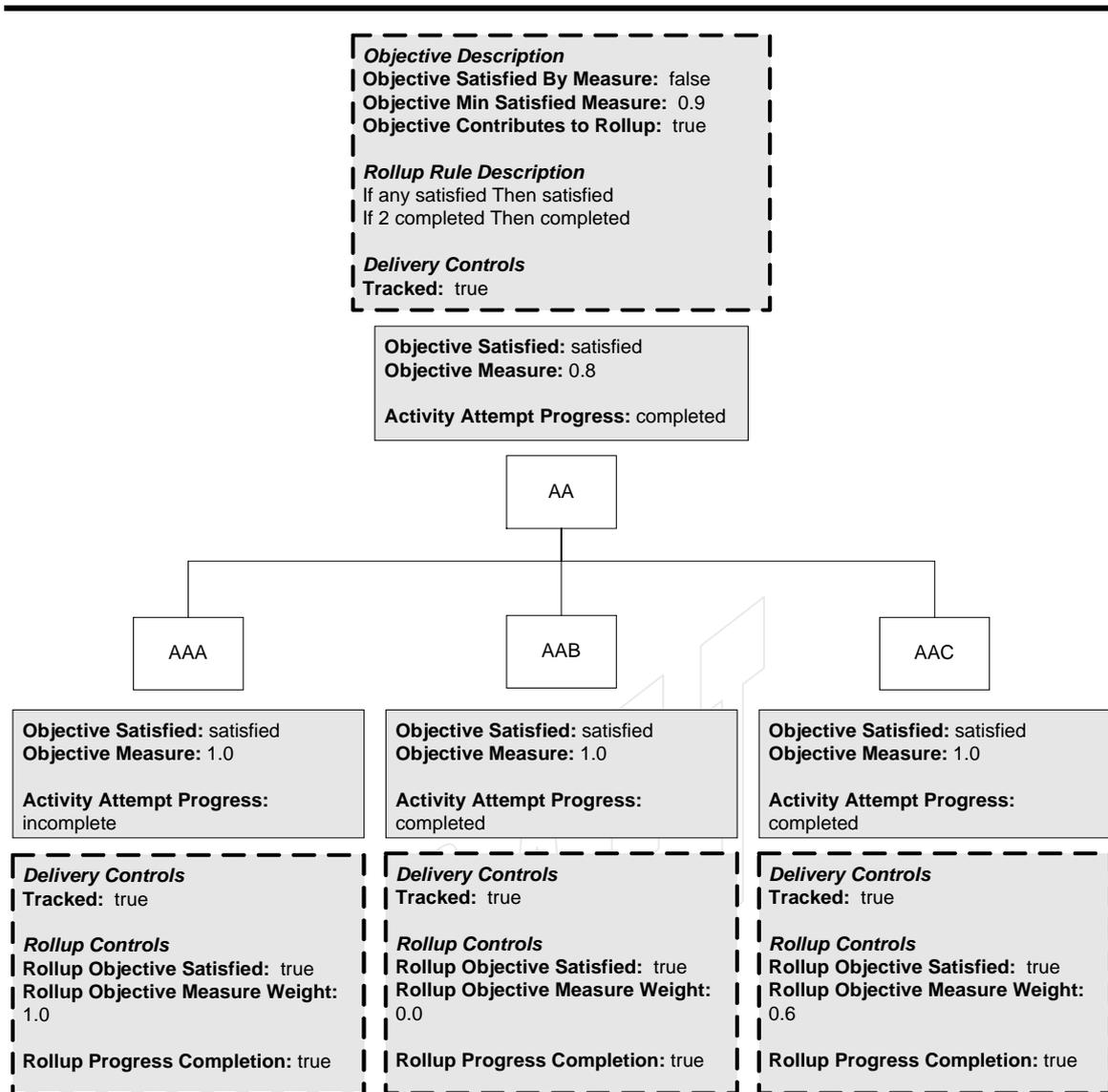


図7.1.5.5a: ルールを使ったアクティビティ進捗状態ロールアップ

2. 既定ルール – もしアクティビティにアクションが *complete* か *incomplete* のルールが定義されていなければ, 以下の既定のロールアップルールが評価される .
 - If all completed, Then completed: すべての子アクティビティが完了なら完了
 - If all (attempted or incomplete), Then incomplete: すべての子アクティビティが実行されるか非完了なら非完了

既定ルールはロールアップルールが評価されるのと同じように評価される – つまりアクションが *incomplete* のルールが最初に評価される .

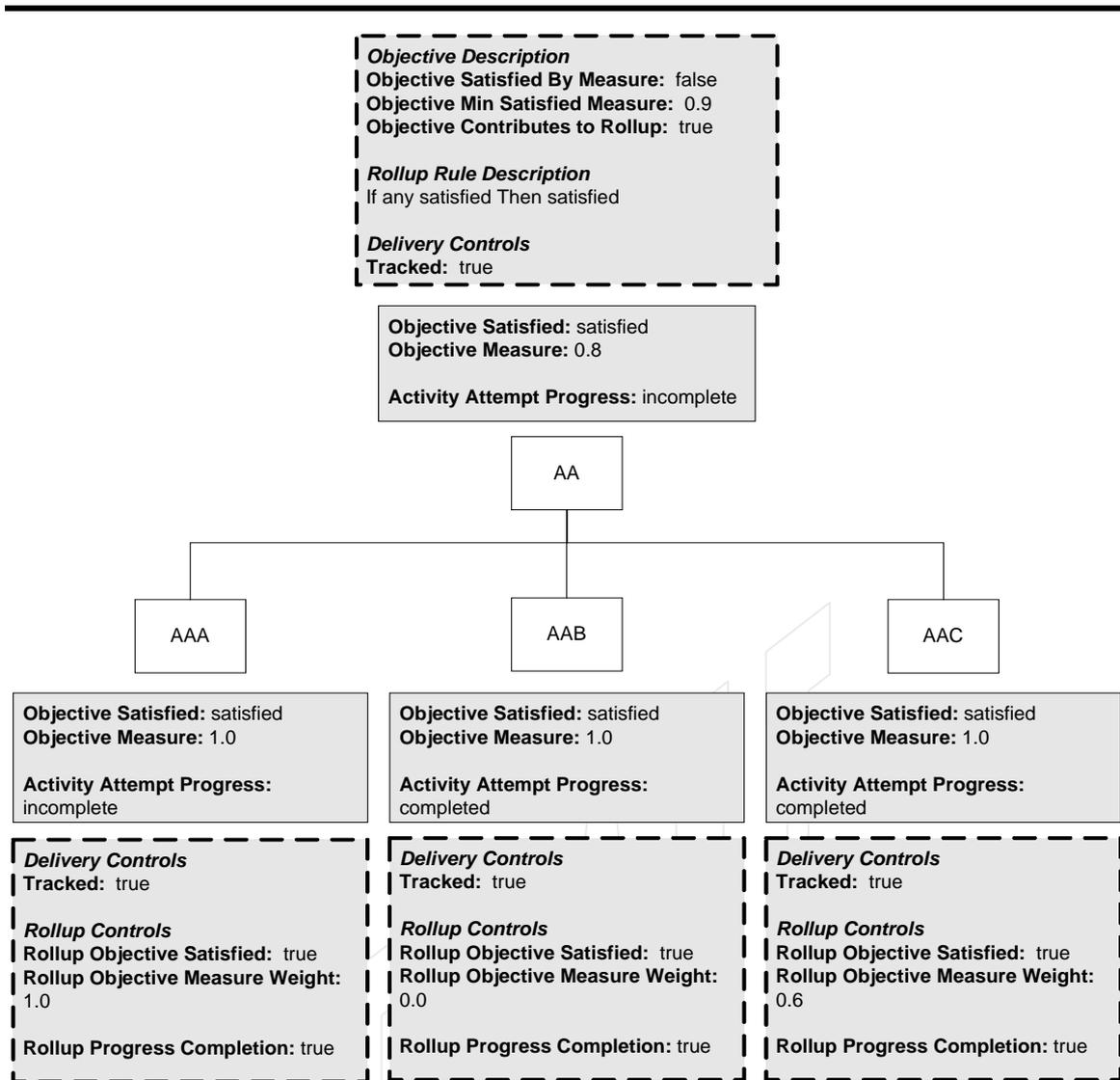


図7.1.5.5b: 既定ルールを使ったアクティビティ進捗ロールアップ

アクティビティ進捗ロールアッププロセスはクラスタのアクティビティ試行完結度に影響しない。LMSのシーケンシングエンジンはアクティビティ試行完結度の値を管理も使用もしない。学習目標ロールアップおよびアクティビティ進捗ロールアップの動作に影響を与えない限り、LMSはアクティビティ試行完結度に対するロールアップ動作の拡張を自由に定義・実装できる。

7.1.6. 選択ランダム化動作

この節の情報はIMS SS仕様の選択ランダム化動作の置換えでなく補足を意図している。詳細についてはIMS SS仕様を参照されたい。

選択ランダム化動作は、いつクラスタの子の一部(ないし全て)を選択し並べ替えるかについて述べている。これらのプロセスは、様々なシーケンシングプ

ロセス（シーケンシング動作 7.1.7 参照）において利用・参照されるアクティビティに影響を与える。

オーバーオールロールアッププロセスは、終了要求が処理された後で選択ランダム化プロセスを起動する、としているが、これは完全に正確ではない。子アクティビティ選択および子アクティビティランダム化プロセスは、LMS シーケンシングエンジンにより適切に起動されることを意図しており、アクティビティの選択ランダム化コントロール（5.1.9 選択ランダム化コントロール参照）に定義されているように、終了動作、シーケンシング動作、配信動作のいずれかで起動される。さらに、LMS がナビゲーション要求の仮の評価を行っている場合、選択ランダム化プロセスはオーバーオールシーケンシングプロセスの外側で起動される場合もある。SCORM 準拠 LMS に対する唯一の要求は、関連するシーケンシング定義モデル要素のタイミング属性と合致するように、選択ランダム化プロセスを適用することである。

- **Never** – Never は選択およびランダム化プロセスで用いられる。全ての子アクティビティは常に作成時の順序で参照される。
- **Once** – 選択およびランダム化プロセスを現在のシーケンシングセッションで一度だけ用いる。これは子アクティビティを他のシーケンシング動作プロセスで参照する以前に行われる。LMS は、通常、シーケンシングセッションを開始する前のタイミングで、全てのアクティビティに対して選択およびランダム化プロセスを適用する。
- **On Each New Attempt** – アクティビティの新しい試行の間ないし直前に、選択およびランダム化プロセスを適用する。利用可能な子アクティビティの正しく一貫した集合がロールアップその他のシーケンシング動作プロセスで利用されるように、LMS は、通常、アクティビティに対する最初の試行が始まる前、および、試行が終わった直後に（試行終了プロセスで）、アクティビティに選択およびランダム化プロセスを適用する。

7.1.6.1 子アクティビティ選択プロセス

選択プロセスにより、コンテンツ作成者は、特定の学習戦略で要求されるよりも多くのアクティビティをクラスタに含めることができる。これにより、異なる学習者が異なる学習アクティビティを経験することが可能になる。コンテンツ作成者は、クラスタのアクティビティの一部が学習者に提示されるように定義することができる。選択プロセスは決められた個数の子アクティビティを、相対的な順番を保って選択する。選択されたアクティビティのみが様々なシーケンシングプロセスで参照される。

7.1.6.2 子アクティビティランダム化プロセス

ランダム化プロセスにより、コンテンツ作成者は、特定の学習戦略に合致するように、学習者が経験するアクティビティの順番を変更することができる。こ

れにより，異なる学習者が同じ学習資源の集合を異なる順番で経験することが可能になる．コンテンツ作成者は，クラスタの利用可能なアクティビティ（コンテンツ作成者が定義したもの，ないし，子アクティビティ選択プロセスで選択されたもの）がランダムに再順序付けされるよう定義することができる．ランダム化プロセスは，どのアクティビティが利用可能なアクティビティの集合に含まれるのかは変更せず，単に順序を変えるだけである．様々なシーケンシングプロセスにおいて，子アクティビティは，ランダム化プロセスで決められた順番で参照される．

7.1.7. シーケンシング動作

この節の情報は IMS SS 仕様のシーケンシング動作の置換えでなく補足を意図している．詳細については IMS SS 仕様を参照されたい．

シーケンシング動作は IMS SS 仕様で不可欠なものである．シーケンシング動作の目的は，現在のアクティビティツリーの状態において *Current Activity* から定義された方法でアクティビティツリーを辿って‘次に’配信するアクティビティを決める，あるいは，新しいシーケンシングセッションを開始して最初に学習者に配信するアクティビティを決定することである．

シーケンシングプロセスはアクティビティツリーの状態を変更しない．*Current Activity* を変更せず，アクティビティのトラッキング状態情報にも影響しない．シーケンシング動作は，起動された時点のアクティビティツリーの状態が現行のものである，と想定する．

シーケンシング動作が，オーバオールシーケンシングプロセスの一部として起動されたとき，配信するアクティビティを決定できない可能性がある．この状況をうまく取り扱い学習者の経験を適切に管理することは，LMS に委ねられている．

7.1.7.1 シーケンシング要求

SCORM 準拠 LMS は，以下のシーケンシング要求を受け付け，対応する動作を実行しなくてはならない(表 7.1.7.1a)

シーケンシング要求	動作
<i>Start</i>	<p>アクティビティツリーのルートからシーケンシングプロセスを開始する．前方順序付きツリートラバーサルを行いアクティビティの評価を行う</p> <ul style="list-style-type: none"> • アクティビティが“Flow”シーケンシング制御モードを持たない(値が <i>False</i>)場合，配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ． • ツリートラバーサルが葉アクティビティに到達したら，そのアクティビティが学習者への配信候補となる．

Resume All	<p>先立つSuspend All終了要求が処理されたアクティビティを再開する。再開するアクティビティはSuspended Activityのアクティビティ状態属性に保持されている。</p> <ul style="list-style-type: none"> もしSuspended Activityのアクティビティ状態属性の値が定義されたアクティビティであれば、そのアクティビティ（およびその全ての祖先のアクティビティ）に対する試行が再開されなくてはならない、そしてSuspended Activityが学習者への配信候補となる。 もしSuspended Activityのアクティビティ状態属性の値が未定義なら、配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ。
Continue	<p>Continueシーケンシング要求は、現アクティビティの親の“Flow”シーケンシング制御モードが実行可(値が“True”)の場合のみ処理される。それ以外の場合、このシーケンシング要求は何も影響を持たない。</p> <p>現アクティビティからシーケンシングプロセスを開始する。前方順序付きツリートラバーサルを行いアクティビティの評価を行う</p> <ul style="list-style-type: none"> アクティビティが“Flow”シーケンシング制御モードを持たない(値が“False”)場合、配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ。 葉アクティビティに到達したら、そのアクティビティが学習者への配信候補となる。
Previous	<p>Previousシーケンシング要求は、現アクティビティの親の“Flow”シーケンシング制御モードが実行可(値が“True”)の場合のみ処理される。それ以外の場合、このシーケンシング要求は何も影響を持たない。</p> <p>Previousシーケンシング要求は、現アクティビティの“Forward Only”シーケンシング制御モードが“False”の場合のみ処理される。</p> <p>現アクティビティからシーケンシングプロセスを開始する。後方順序付きツリートラバーサルを行いアクティビティの評価を行う</p> <ul style="list-style-type: none"> アクティビティが“Flow”シーケンシング制御モードを持たない(値が“False”)場合、配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ。 葉アクティビティに到達したら、そのアクティビティが学習者への配信候補となる。
Choice	<p>このシーケンシング要求には、要求の“ターゲット”，すなわち次に配信するものとして要求されたアクティビティが付随する。</p> <p>現アクティビティの親のアクティビティが“Choice”シーケンシング制御モード(値が“True”)を持たない場合、配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ。</p> <p>現アクティビティ（あるいは現アクティビティが不定の場合ルートアクティビティ）とターゲットアクティビティの共通の祖先アクティビティを見つける。現アクティビティから共通の祖先アクティビティへ、そしてターゲットアクティビティへ経路を作る。</p> <p>現アクティビティの親から共通の祖先アクティビティへの経路上で、ただし共通の祖先を含まない各アクティビティが“Choice Exit”シーケンシング制御モード(値が“True”)を持たない場合、配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ。</p> <p>現アクティビティからシーケンシングプロセスを開始する。現アクティビティからターゲットアクティビティへの経路を辿りアクティビティを評価する（経路を辿る各ステップは、順序付きツリートラバーサルに対して相対的に前方ないし後方となる）。</p> <ul style="list-style-type: none"> もし前方に辿る場合は、アクティビティに対してStop Forward Traversalルールが“True”なら、配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ。 もし後方に辿る場合は、アクティビティに対してForward Onlyルールが“True”なら、配信するアクティビティを決定せず終了する - シーケンシングシステムは他のシーケンシング要求を待つ。 ターゲットアクティビティに到達したら、ターゲットアクティビティが学習者への配信候補となる。
Retry	<p>現アクティビティが配信候補となる（新しい試行として記録される）。</p>
Exit	<p>現アクティビティからシーケンシングプロセスを開始する。</p> <ul style="list-style-type: none"> 現アクティビティがアクティビティツリーのルートならシーケンシングを終了する。それ以外の場合、シーケンシングシステムは他のシーケンシング要求を待つ。

表 7.1.7.1a シーケンシング要求の定義

シーケンシング要求はその動作により 4 つのカテゴリに分類される。

- **シーケンシングセッションを開始する** – *Start, Resume All*, (シーケンシングセッション開始前の) *Choice* – これらの要求では *Current Activity* が未定義である必要がある (シーケンシングセッションがまだ始まっていない)。新しいシーケンシングセッションで学習者が経験する最初のアクティビティを決定しようとする。

注意：配信アクティビティの決定に成功しても、そのアクティビティが実際に配信されるかどうかは保証されない (7.1.8 配信動作参照)。最初のアクティビティが学習者に完全に配信されるまで、シーケンシングセッションが開始したとは見做されない。

- **アクティビティツリーを“次の”アクティビティに向かって迎る** – *Continue, Previous*, (シーケンシングセッション開始後の) *Choice* – これらの要求では *Current Activity* が定義されている必要がある (シーケンシングセッションがすでに始まっている)。「次の」配信アクティビティを見つけるために、*Current Activity* からアクティビティツリーを決められた方法で迎る。
- **現アクティビティを繰り返す** – *Retry* – この要求では *Current Activity* が定義されている必要がある (シーケンシングセッションがすでに始まっている)。*Current Activity*, ないし、*Current Activity* がクラスタの場合はその最初の配信可能な子アクティビティを配信しようとする。
- **シーケンシングセッションを終了する** – *Exit* – この要求では *Current Activity* が定義されている必要がある (シーケンシングセッションがすでに始まっている)。*Current Activity* がアクティビティツリーのルートの場合、シーケンシングセッションは終了する – オーバオールシーケンシングプロセスは終了し、制御は LMS に戻る。*Current Activity* がアクティビティツリーのルートでない場合、配信アクティビティは決定されず LMS のシーケンシングエンジンは他のナビゲーション要求が発行されるのを待つ。

7.1.7.2 シーケンシング要求プロセス

オーバオールシーケンシングプロセスは、ナビゲーション動作ないし終了動作からのシーケンシング要求を伴ってシーケンシング要求プロセスを起動する。シーケンシング要求プロセスの結果、学習者に配信する“次の”アクティビティが決定される。シーケンシング要求プロセスは、シーケンシング要求に基づき適切なシーケンシングプロセスを起動して処理を実行する。全てのシーケンシングプロセスは、*Current Activity* が未定義の時に実行されるプロセスも含め、*Current Activity* から処理を開始する。

オーバーオールシーケンシングプロセスの範囲外でシーケンシング要求プロセスを起動するような実装も可能である。さらに、様々なシーケンシングプロセスを *Current Activity* 以外のアクティビティから起動し追加的な例外情報を記録して、“知的な” UI (ユーザインターフェース) 制御やシーケンシング例外のコンテキストを可能とすることもできる。SCORM 準拠の LMS は、オーバーオールシーケンシングプロセスのコンテキスト内でシーケンシング要求プロセスを起動した場合、IMS SS 仕様に記述された動作を行わなくてはならない。

7.1.7.3 制限条件の評価

コンテンツ作成者はアクティビティの利用の制限を定義できる。これらの制限は、最大試行回数、時間制御、利用期限である。制限条件はシーケンシング定義モデル (5章参照) で定義されている。様々なシーケンシングプロセスにおいて、定義された制限条件に反するアクティビティは配信されない。さらに、制限条件に反するアクティビティは‘無効化’され、兄弟や子孫のアクティビティの配信は考慮されなくなる。

制限条件の評価は制限条件チェックプロセスで (IMS SS 仕様で定義されたように) 行われる。

7.1.7.4 事前条件(precondition)シーケンシングルールの評価

アクティビティはひとつないし複数の事前条件シーケンシングルールを有している場合がある。

シーケンシングルールは以下の構造を有する。

If condition set Then action

condition set はアクティビティのトラッキング状態情報に対して評価される条件の集合である。もし *condition set* が True ならば *action* が適用される。

シーケンシングルールの動作(action)は、評価のタイミング、どのシーケンシングプロセスに適用されるか、これらのプロセスへの影響、に応じて、3つの種別に分けられる。事前条件シーケンシングルールは、様々なシーケンシングプロセスで様々な時点で評価される。

事前条件シーケンシングルールはシーケンシングルールチェックプロセスで評価される。

以下に例を示す：

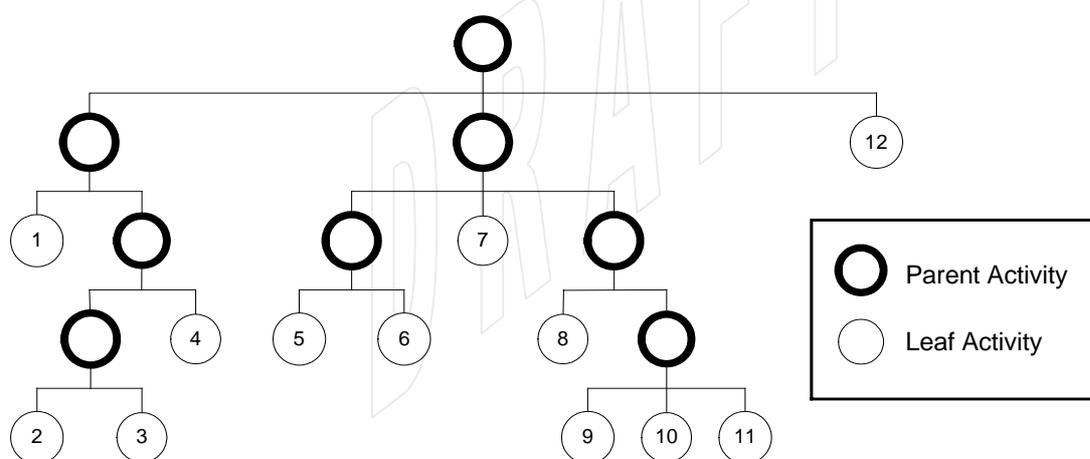
- **If satisfied Then skip** – もしアクティビティが習得されていれば、フロープロセスにおいてアクティビティを飛び越す。
- **If outside available time range Then hide from choice** – もし現時刻がアクティビティの利用可能時間範囲外ならば、アクティビティを choice シーケンシング要求のターゲットとしない。

- **If attempted Then disable** – もしアクティビティが実行されていたら，アクティビティを無効とする．
- **If incomplete Then stop forward traversal** – もしアクティビティが未完了ならば，シーケンシングプロセスの間，（前方の）兄弟アクティビティへの参照を禁ずる．

上の例は，アクティビティに定義可能なシーケンシングルールの一部のタイプを示している．

7.1.7.5 フローサブプロセス

フローサブプロセスは，LMS のシーケンシングエンジンが，アクティビティツリーを決められたアクティビティから決められた方向にどのように辿るのかを定義する．フローサブプロセスは多くのシーケンシングプロセス(*Start, Retry, Choice, Continue, Previous*)で，LMS のシーケンシングエンジンがアクティビティツリーを辿る動作を制御する際に使用される．フローサブプロセスは葉アクティビティでのみ停止し（配信を行おうと）する．フローサブプロセスは，シーケンシング制御モード Flow が False に設定されたアクティビティに行き当たると失敗する．以下の図は，ツリー全体に対して Flow が有効であるとしたときの，アクティビティツリーの葉の相対的な順番を示している．



フロープロセスは以下のように要約できる．

1. 指定されたアクティビティから指定された方向へ 1 アクティビティの移動を試みる．
- Loop**
2. アクティビティが無効でないことを確かめる．無効ならフローサブプロセスを終了する．
 3. アクティビティが制限条件に反していないことを確かめる．反していればフローサブプロセスを終了する．

-
4. もしアクティビティを飛び越す場合，指定されたアクティビティから指定された方向へ1アクティビティの移動を試みる．ステップ2へ戻る．
 5. もしアクティビティが葉ならば，アクティビティは配信候補となり，フローサブプロセスを終了する．
 6. 適切な方向でクラスタに入る．
 - もし前方に移動する場合，次のアクティビティはクラスタの最初の子である．
 - もし後方に移動する場合，Forward Only が False ならば，次のアクティビティはクラスタの最後の子である．
 - もし後方に移動する場合，Forward Only が True ならば，次のアクティビティはクラスタの最初の子である – (クラスタの中で) 一時的に前方に移動する．
 7. アクティビティが決まらなければ，フローサブプロセスを終了する．
 8. ステップ2へ戻る．

7.1.7.6 シーケンシングプロセス

この節の情報は IMS SS 仕様のシーケンシング要求プロセスで詳述されている動作の置換えでなく補足を意図している．詳細については IMS SS 仕様を参照されたい．

7.1.7.6.1 Start シーケンシング要求プロセス

Start シーケンシング要求プロセスでは，シーケンシングセッションがまだ開始されていない必要がある．新しいシーケンシングセッションを開始しようとして，アクティビティツリーを移動する．このプロセスはフローサブプロセスを利用する（7.1.7.5 参照）．

シーケンス制御モード Flow が False に設定されたアクティビティに行き当たってフローサブプロセスが終了した場合，何も配信されずシーケンシングセッションは始まらない．

7.1.7.6.2 Resume All シーケンシング要求プロセス

Resume all シーケンシング要求プロセスでは，シーケンシングセッションがまだ開始されていない必要がある．*Suspended Activity* のアクティビティ状態属性を見て，以前のセッションをどこから再開するか決定する．

Suspended Activity が定義されていなければ，プロセスは終了する．*Suspended Activity* が定義されていれば，ルートから *Suspended Activity* までの全アクティビティの試行が再開され，シーケンシングセッションを開始し，*Suspended Activity* が学習者に配信される．

7.1.7.6.3 Retry シーケンシング要求プロセス

Retry シーケンシング要求プロセスは、終了プロセスで事後条件ルールを評価することによって起動される。このプロセスは、シーケンシングセッションが開始されていて、*Current Activity* が再試行のターゲットであることを前提とする。もし *Current Activity* がクラスタなら、retry プロセスはフローサブプロセス（7.1.7.5 参照）を起動し、学習者が次に経験するアクティビティを決定する。アクティビティの再開により新しい試行が始まる。

7.1.7.6.4 Exit シーケンシング要求プロセス

Exit シーケンシング要求プロセスはシーケンシングセッションが開始されていることを前提とする。配信要求は返却されない。*Current Activity* がアクティビティツリーのルートなら、シーケンシングセッションを終了し、制御を LMS に戻す。

7.1.7.6.5 Continue シーケンシング要求プロセス

Continue シーケンシング要求プロセスはシーケンシングセッションが開始されていることを前提とする。開始すると、*Current Activity* から前方へのフローサブプロセス（7.1.7.5 参照）を起動する。フローサブプロセスがアクティビティを決定すると、そのアクティビティを配信要求に設定する。

7.1.7.6.6 Previous シーケンシング要求プロセス

Previous シーケンシング要求プロセスはシーケンシングセッションが開始されていることを前提とする。開始すると、*Current Activity* から後方へのフローサブプロセス（7.1.7.5 参照）を起動する。フローサブプロセスがアクティビティを決定すると、そのアクティビティを配信要求に設定する。

7.1.7.6.7 Choose シーケンシング要求プロセス

Choose シーケンシング要求プロセスは学習者（ないしシステム）が指定したアクティビティを配信候補とする。シーケンシングセッションが既に始まっている場合、choice プロセスはアクティビティツリーを *Current Activity* からターゲットアクティビティへ辿ろうとする。シーケンシングセッションがまだ始まっていない場合、choice プロセスはアクティビティツリーをルートからターゲットアクティビティへ辿ろうとする。choice プロセスがアクティビティを決定し、それが葉で無い場合、そのアクティビティから前方へのフローサブプロセス（7.1.7.5 参照）を起動する。葉アクティビティが指定されるとそれが配信される。

7.1.8. 配信動作

この節の情報は IMS SS 仕様の配信動作の置換えでなく補足を意図している。詳細については IMS SS 仕様を参照されたい。

配信動作はオーバーオールシーケンシングプロセスの最後のステップを定義している。配信動作の目的は、指定された配信要求について、その要求の有効性を

確認し、有効であれば適切な学習資源を配信することである。LMSは、コンテンツパッケージを使用して、指定されたアクティビティに関連した配信学習資源を決定する。もし、配信動作がオーバーオールシーケンシングプロセスの一部として起動された場合、配信要求が有効とならない可能性がある。この状況をうまく取り扱い学習者の経験を適切に管理することは、LMSに委ねられている。

配信動作（配信要求プロセス）は、オーバーオールシーケンシングプロセスのコンテンツ外でLMSによって起動される可能性がある。これは、可能性のある配信要求の先行評価を行うために実行される。配信要求プロセスはトラッキング情報に影響を与えないので、副作用を意識せずに起動できる。しかし、どのような結果であれ、適切な扱いをすることは実装の責任である。

SCORMのひとつの目標は、学習資源が複数のLMSに渡って再利用・相互運用可能であることである。これを可能とするために、学習資源の実行を開始する共通の方法がなくてはならない。コンテンツ配信環境は、LMSのシーケンシングエンジンとSCORM配信機構の橋渡しを規定する。それは、学習資源の配信を予定するアクティビティツリーの状態を管理し、SCORM配信機構にその学習資源を示す。

SCORM配信機構はLMSがWebベースの学習資源の実行を開始する共通の方法を定義する。この仕組みは、配信された学習資源とLMSの間の通信の確立の手順と役割分担を定義している。通信プロトコルは共通APIを使用することによって標準化されている。

共通配信スキームは、表面下のLMSの実装を指定せずに複数LMSに渡る学習資源配信動作の一貫性を可能とする。この文脈で、LMSという用語は、学習資源の配信を管理する機能を有するシステムを指すことに注意されたい。この配信スキームは、学習経験においてSCOないしSCAという形式でのWebを用いた学習資源の配信について述べている。

7.1.8.1 配信要求プロセス

配信要求プロセスは、配信要求で指定されたアクティビティの配信が可能かどうか決定する。このプロセスは、アクティビティツリーをルートから配信要求で指定されたアクティビティまで辿り、各アクティビティが無効化されていたり制限条件に反していないことを確認する。

7.1.8.2 コンテンツ配信環境プロセス

コンテンツ配信環境プロセスは、オーバーオールシーケンシングプロセスが起動する最後のプロセスである。有効と仮定された配信要求を受け取り、指定されたアクティビティを配信するようアクティビティツリーの準備をする。このプロセスは以下を含む。

-
1. 指定されたアクティビティを配信する際にアクティブとならない全てのアクティビティの試行を終了する。
 2. 指定されたアクティビティを配信する際にアクティブとなる非活性アクティビティの全ての試行を開始（または再開）する。
 3. 全てのアクティブなアクティビティのトラッキング情報を適切に初期化する。
 4. 配信アクティビティおよび関連する補助アクティビティを LMS に示す。

コンテンツ配信環境プロセスを終了する際、LMS のシーケンシングエンジンは LMS に制御を戻し、他のナビゲーション要求を待つ。

コンテンツ配信環境プロセスは、オーバオールシーケンシングプロセスの範囲外では起動できない。起動すると非準拠動作を起こす可能性がある。

7.1.8.3 SCORM 配信機構

7.1.8.3.1 試行の開始

配信すべき適切な学習アクティビティを決定すると、LMS はアクティビティに対応した起動すべき学習資源が存在するか否か決定する。もし、学習アクティビティが起動可能な学習資源を参照していれば、LMS はコンテンツパッケージマニフェストに定義された学習資源の起動ロケーションを定義した URL を使って、現在表示されている学習資源を置き換え、起動ロケーションで見つかる学習資源に導く。

LMS は学習資源の起動をどのように実装しても良く、必要に応じて実際の起動の役割を LMS のクライアントないしサーバ部分に委譲しても良い。実際の起動は HTTP プロトコルを使って実行しなくてはならない。最終的には、コンテンツパッケージの起動ロケーションによって指定される学習資源は、クライアントのブラウザで起動・配信される。

7.1.8.3.2 学習資源起動の準備

指定された学習資源を起動する準備として、LMS は以下の機能を実行しなくてはならない。

1. 学習資源が起動パラメータを必要とするか決定する。
2. 学習資源を起動する絶対 URL を決定する。
3. ランタイム状態モデルの適切な部分を初期化する。
 - a. 学習資源に対応したアクティビティのトラッキングモデルの項目を初期化する。
 - b. 学習資源に対応した SCORM ランタイム環境データモデルの要素を初期化する（もし学習資源が SCO ならば）。

7.1.8.3.3 学習資源の起動

SCORM は学習アクティビティに対応する学習資源として二つのタイプを規定している。Sharable Content Object (SCO, 共有コンテンツオブジェクト) は, LMS と通信するための SCORM ランタイム環境を利用する固有の起動可能アセットを含む, ひとつないし複数のアセットの集合を表す。SCO は, SCORM ランタイム環境を利用して LMS が記録可能な学習資源の最小単位を表す。Sharable Content Asset (SCA, 共有コンテンツアセット) は, 固有の起動可能アセットを含むひとつないし複数のアセットの集合を表す。SCA は SCO と異なり LMS との (API を経由した) 通信を行わない。

7.1.8.3.3.1 SCA の起動

アセットを表す学習資源に対して SCORM 起動モデルが要求するのは, LMS がアセットを HTTP プロトコルを使って起動するという点だけである。アセットは API やデータモデルを使って LMS と通信する必要が無いので, アセットは LMS が提供する API アダプタを見つける必要は無い。

7.1.8.3.3.2 SCO(Sharable Content Objects)の起動

SCO を表す学習資源に対して SCORM 起動モデルが要求するのは, LMS が一度にひとつの SCO だけを起動し, 一度にひとつの SCO だけがアクティブになるということである。SCO は他の SCO を起動してはならない。

LMS は, Document Object Model (DOM)オブジェクトとして API アダプタをアクセス可能とする LMS ウィンドウの子ウィンドウないし子フレームのブラウザウィンドウ内で SCO を起動する。

SCO は, API アダプタが見つかるまで親ないしオープンウィンドウを再帰的に検索しなくてはならない。API アダプタが見つかったら, SCO は LMS との通信を開始できる。

このページは空白である。

DRAFT

8 章

ナビゲーションユーザインタフェース 管理

DRAFT

このページは空白である。

DRAFT

8.1. ユーザインタフェース管理の背景

SCORM コンテンツは学習経験を提供するものである。ランタイムサービス (RTS)はコンテンツを配信し、(学習)経験を管理する。RTS は LMS の一部あるいは、オンラインまたはオフラインのサービスであり、LMS に替わってコンテンツを配信する。

RTS は SCORM1.3 の範疇に入る 2 つのタイプのユーザインタフェースイベントを管理する。ひとつはナビゲーションイベントであり、他のイベントは広義には補助サービス(auxiliary service)に分類されるもので、直接的に SCORM コンテンツに関連するものである。

ナビゲーションサービスと補助サービスは、そのイベントを始動させるユーザインタフェース装置があることを前提にしている。学習者がそういう装置を始動させると、RTS はイベントに対応する要求に変換し、要求を処理し、その要求が有効であるかどうかおよびどのようなアクションをとるかを決定し、その後、当該のアクションを実行する。例えば、パッケージが用語集(Glossary)と識別される補助リソースを含んでいる場合、ユーザがツールバーで対応するボタンをクリックすると RTS は用語集を立ち上げる。

SCORM V1.3 ではユーザインタフェースの型、様式については、ナビゲーションや補助サービス用のユーザインタフェース装置を含めて、何の要求もしない。ユーザインタフェースの性質および学習者と RTS 間のやりとりのメカニズムは意図的に規定されていない。ルック&フィール、表現スタイル、ユーザインタフェース装置の位置や制御手段などは SCORM V1.3 の規格の範囲外である。

8.2. ナビゲーションの背景

SCORM V1.3(の文脈)では、ユーザに与えられる学習経験は学習アクティビティの集合としてサポートされる。学習アクティビティは、コンテンツパッケージマニフェスト中のアイテムに対応し、任意の深さにネストできる。それは、要求される学習分類、あるいは分解可能なアクティビティの複雑さに依存している。ネストの最下層のレベルの学習アクティビティ(葉アクティビティ)は関連する学習リソースを規定する。各学習リソースはコンテンツパッケージの中では、“Resource”で表わされる。“Resource”は、典型的には、コンテンツパッケージマニフェストの中で、“File”で識別される種々のアセットを持つ。あるいは、“Resource”は外部サービスから提供されるシミュレーションのように、パッケージ中に含まれない SCO や SCA を参照してもよい。アクティビティは、学習経験を提供するために起動される起動可能なリソースをひとつだけ指定するが、他のアクティビティも同じリソースを参照してよい。

SCORM V1.3 は 2 つのタイプの学習リソースを認識し、それらのリソースは学習アクティビティが提供されるときに起動される。すなわち、

- 単一の起動可能なリソースとしてパッケージされたひとつ以上のアセットからなる SCO で、SCORM のランタイム環境 API を通じて LMS と通信する。
- 単一の起動可能なリソースとしてパッケージされたひとつ以上のアセットまたはサービスからなる SCA で、SCORM のランタイム環境 API を通じて LMS とは通信しない。これらは非通信コンテンツオブジェクトと呼ばれる。

SCO や SCA は SCORM ランタイム環境の起動要求に適合しなければならない。言い換えれば、RTS から起動された SCO や SCA は、その同一ウィンドウ上で置き換わる他の SCO や SCA を起動してはならない。

8.2.1. ナビゲーションの概観

SCORM では、ナビゲーションは、学習アクティビティを通して学習者と RTS が協調して特定のパスを定義するプロセスである。コンテンツアグリゲーションの作者が定める範囲内で、学習者はナビゲーションリクエストを表すユーザインタフェース装置を始動させることができる。これらのユーザインタフェース装置は RTS または現在のアクティビティで使用されているコンテンツオブジェクトあるいはその両方から提供される。

学習者がナビゲーションインタフェース装置をとおしてナビゲーションイベントを発生させた場合、RTS はイベントを処理し、次に何をするかを決定する。何かとは以下に示す中のひとつである。

- ナビゲーションリクエストの意味 (effect) がコンテンツアグリゲーションの現在の試行の終了である場合、RTS はコンテンツアグリゲーションのルートアクティビティの試行を終了させる“Exit All”ナビゲーション要求を処理して LMS へ制御を戻す。
- 現トラッキング状態と適用規則を評価した後、RTS は(ナビゲーションイベントに基づく)意図したナビゲーションリクエストの処理は認められないと決定する。この場合、RTS はリクエストを拒否し、イベントは無視される。RTS は、学習者が他のナビゲーションイベントを発生するまで何らの継続アクションもとらない。
- 現在のトラッキング状態と適用規則を評価した後、RTS は (ナビゲーションイベントに基づく) 意図したナビゲーションリクエストの処理は認められると決定する。RTS は対応するナビゲーションリクエストを処理し、シーケンシングエンジンによって配信のために特定されたアクティビティに関連する学習リソースを起動する。

SCORM シーケンシングの適用範囲はアクティビティ間のナビゲーションに対してのみである。アクティビティはマニフェスト中の“Item”で表現され、起動可能なリソース (SCO または SCA) によりサポートされたり、されなかったりする。この時、SCORM は、単一アクティビティ (Item) に対応する SCO または SCA の内部のシーケンシングやナビゲーションを定義する力を持たない。例えば、SCORM ナビゲーションは複数ページ SCO の個々のページ間には適用されず、ある SCO から他の SCO へのナビゲーションに適用される。IMS のシンプルシーケンシング仕様は、ユーザインタフェース装置をとおして学習者によって発生されるナビゲーションイベントの集合を定義している。これらのイベントは LMS または SCO から提供されるユーザインタフェース装置からも発生することができる。SCO 内でイベントがどのように発生されるかについては SCORM では定義されていない。例えば、SCO 内の発生イベントがユーザアクションであるのか、あるいは内部に記述されたイベントの結果であるのかは SCORM の範囲外である。

SCORM で定義されるナビゲーションイベントにより、学習者は一連の学習アクティビティをナビゲートすることができる。これは以下のような希望するアクションを指定することによって行える、例えば、ある特定の学習アクティビティを選択する、次のアクティビティに継続する、前のアクティビティに戻る、などである。その他のナビゲーションイベントでは、学習者が学習アクティビティを終了したり、学習アクティビティの試行をやめたりすることができる。

8.2.2. ナビゲーションイベント

SCORM では、すべてのナビゲーションイベントは、LMS の一部もしくはコンテンツを配信するために LMS から起動されるサービスである RTS によって処理される。ナビゲーションイベントに基づいて RTS は対応する相応なアクションを決定し、それを実行する。例えば、学習者はある学習アクティビティを選ぶためにメニューを使うか、木構造のユーザインタフェース制御を使用する。RTS は対応するナビゲーションイベントを処理し、そして、コンテンツアグリゲーション、トラッキング状態モデルおよび対応するシーケンシング情報に基づいて、選ばれたアクティビティに関連するコンテンツオブジェクトを起動する。同様に、学習者は SCO から提供された“次”ボタンをクリックすることができる。SCO は、次に対応する相応なナビゲーションイベントを RTS に対し通信しなければならない。RTS は、これによって、次にどのコンテンツオブジェクトを起動するか、コンテンツアグリゲーションおよび対応するシーケンシング規則に基づいて決定することができる。

下記の表は、一連のナビゲーションイベント定義するとともに、これらのナビゲーションイベントがどのようにナビゲーション要求に対応するかを定義している。更に、この表は各ナビゲーションイベントの発生源を定義している。すべてのイベントは RTS から発生される一方、いくつかのイベントは SCO から発生される。

SCORM V1.3 ではユーザインタフェースの型，様式については，ナビゲーション用のユーザインタフェース装置を含めて，何の要求もしない．ユーザインタフェースの性質および学習者と RTS 間のやりとりのメカニズムは意図的に規定されていない．ルックアンドフィール，表現スタイル，およびナビゲーション制御の位置などは SCORM V1.3 の規格の範囲外である．

RTS が適合となるためには，RTS はナビゲーションイベントを管理しなければならず，また各ナビゲーションイベントへの応答には下記の表に定義されている動作を示さなければならない．

ナビゲーションイベント	動作記述	発信源
Start	このナビゲーションイベントは “ <i>Start</i> ” ナビゲーション要求を発生させる．このイベントは典型的には，学習者がコンテンツアグリゲーションを起動した時，自動的に RTS によって生成される．	RTS のみ
Continue	このナビゲーションイベントは “ <i>Continue</i> ” ナビゲーション要求を発生させる．	RTS または SCO
Previous	このナビゲーションイベントは “ <i>Previous</i> ” ナビゲーション要求を発生させる．	RTS または SCO
Choose	このナビゲーションイベントは “ <i>Choose</i> ” ナビゲーション要求を発生させ，このイベントには選択されたアクティビティの識別子が付いている．もし識別子が付与されていないか，もしくは識別子が現アグリゲーションで許されるアクティビティの識別子でない場合，動作は不定となる．	RTS のみ
Abandon	このイベントは，現在渡されたアクティビティおよび対応するコンテンツオブジェクトまたは SCO に対する現在の試行を，後で再試行する用意をせずに，即時終了あるいは異常終了要求を示す． このアクティビティが子アクティビティである場合は，この子アクティビティだけが放棄される．放棄自体は現在渡された親アクティビティには何の影響もない． 放棄された試行は一回の試行に数えられる．放棄は，既に記録された履歴情報(tracking information)を“rolled back”(元に戻す)しなければならないということを意味しない．例えば，アクティビティが既に合格もしくは完了と記録されている場合，それが不合格や非完了にはならない． LMS の実装や方針によっては，現試行中にこのアクティビティに記録される他の履歴情報は保持されるかもしくは捨てられてもかまわない． このイベントは，結果として “ <i>Abandon</i> ” ナビゲーション要求となる．	RTS または SCO
Abandon All	このイベントは，コンテンツアグリゲーション全体に対する現	LMS のみ

	<p>在の試行を，即時終了あるいは異常終了させることを示す．</p> <p>これは，すべての親アクティビティと同様に現在わたされているアクティビティを含む．</p> <p>試行は終了となり，現アクティビティを再試行するような暗黙的な意図は無い．</p> <p>放棄された試行は一回の試行に数えられる．放棄は，既に記録された履歴情報(tracking information)を“rolled back”(元に戻す)しなければならないということを意味しない．例えば，アクティビティが既に合格もしくは完了と記録されている場合，それが不合格や非完了にはならない．</p> <p>LMS の実装や方針によっては，現試行中にこのアクティビティに記録される他の履歴情報は保持されるかもしくは捨てられるかもしれない．</p> <p>このイベントは，結果として“AbandonAll”ナビゲーション要求となる．</p>	
Suspend All	<p>このイベントは，アグリゲーション全体に対する現在試行中の現セッションの終了要求を意味する．</p> <p>アグリゲーション全体でのアクティビティの試行を中断する場合，アグリゲーション全体の状態を再試行するのに必要な状態と他の制御情報は保存しておかなければならない．更に“SuspendAll”リクエストが出された時，いずれかのアクティビティが活動していれば，再試行に必要な情報にはどのアクティビティであるかが含まれていなければならない．LMS は，現学習リソースが SCO であれば， cmi.core.exit の値に“suspended”を設定しなければならない．</p> <p>以前に中断されたアグリゲーション全体を再試行する場合，中断時に活動していたアクティビティが再試行されなければならない．</p> <p>SuspendAll イベントが起きているその時にアクティビティが配信されれば，アクティビティに対する動作は，中断イベントに対するのと同様である．</p> <p>このイベントは，結果として“SuspendAll”ナビゲーション要求となる．</p>	LMS のみ
Unqualified Exit	<p>このイベントは，現在配信されているアクティビティに対する試行が正常に終了し，現在配信されているリソースの終了が他のナビゲーションイベント (Continue , Previous や Choose など) によって始動させられたものではないことを意味する．</p> <p>このナビゲーションイベントは，“Exit”ナビゲーション要求を発生させる．</p>	LMS または SCO
Exit All	<p>このナビゲーションイベントは，“ExitAll”ナビゲーション要求を発生させる．</p>	LMS のみ

表8.2.2a: ナビゲーションイベントとその記述

ナビゲーションイベントは、シーケンシング制御モードに関連して有効化されて発生する。学習アクティビティの親クラスタに対し定義される制御モードは、子クラスタに適用可能なナビゲーションイベントを定義する。choice 制御モードが有効なら、Choose ナビゲーションイベントがクラスタの子アクティビティに適用可能となる。同様に flow 制御モードが有効なら、previous および continue ナビゲーションイベントがクラスタの子アクティビティに対して実行可能となる。

8.3. 補助サービスの背景

SCORM V1.3 では、ユーザに提供される学習経験は学習者アクティビティの集合でサポートされる。上述のように学習アクティビティはコンテンツパッケージングマニフェスト中の *Item* に対応し、任意の深さにネストできる。それは望ましい学習分類あるいは分解可能なアクティビティの複雑さに基づいている。

ネストの最下層の学習アクティビティは対応する学習リソースを定義している。学習アクティビティはさらに、ネストのどのレベルでも、対応する補助リソースを定義できる。それらのリソースは、アクティビティが経験されるときに自動的に提供されない。さらに、補助リソースはナビゲーションイベントをとおして順序付けられない。言い換えれば、アクティビティツリーの葉ノードとして定義される学習リソースのみがナビゲーションイベントから制御される。補助リソースはナビゲーションに影響を与えない補助サービスイベントをとおしてのみ起動される。

SCORM V1.3 は、補助サービスイベントとして使用できる 2 つの学習リソースを認めている。すなわち、

- 単一の起動可能リソースとしてひとつ以上のパッケージングされたアセットからなる SCO で、SCORM のランタイム環境 API をとおして LMS と交信する。
- 単一の起動可能リソースとしてひとつ以上のパッケージングされたアセットまたはサービスからなる SCA で、SCORM のランタイム環境 API をとおして LMS とは交信しない。

補助サービスとリソースは用語集、参照マニュアルなどを含む。ナビゲーションに使用される学習リソースとは異なり、補助サービスはアクティビティのネストレベルとは無関係にどのアクティビティにも定義できる。

8.3.1. 補助サービスの概観

SCORM では、補助サービスは学習者が SCORM パッケージで定義される学習アクティビティをとおしてナビゲートする間に、学習者に提供されるサービスである。各アクティビティ集合の作者によって定義される範囲内で、学習者は現アクティビティのスコープ内で有効な補助サービスを呼び出すインタフェース装置を起動できる。例えば、アクティビティ階層の最上位レベルのアクティビティに対して指定された用語集は、学習者が最上位レベルのアクティビティに含まれる子アクティビティのいずれかにかかわっている時にはいつでも利用できる。

学習者がアクティビティツリーをナビゲートして補助サービスがスコープ内に入った場合、RTS は補助サービスを利用可能にはするが自動的に起動しない。学習者は RTS から提供される対応するユーザインタフェース装置を通して、補助サービスイベントを発生してサービスを起動できる。補助サービスが利用可能なときに起動されるどの SCO も、同様に、対応する補助サービスイベントを発生できる。例えば、SCO 中の何かが RTS に対し用語集が利用可能なときにそれを立ち上げるよう要求できる。

学習者がユーザインタフェース装置をとおして補助サービスイベントを発生させるかもしくはイベントが SCO から発生させられるかした場合、RTS は要求を処理し、サービスを起動するか、また、どのように起動するかを決定する。補助サービスの動作および関連するいかなるユーザインタフェース装置の実際の概観や動作は SCORM 1.3 の範囲外である。特に、SCORM 1.3 は補助サービスの通信や同期化について何も定義しない。

8.4. ユーザインタフェース装置の管理

8.4.1. ナビゲーションのためのユーザインタフェース装置の管理

RTS は最低限、ユーザインタフェース装置を介して、定義されたすべてのナビゲーションイベントを始動させる機能を提供しなければならない。どのようにこれらの装置が実装されるか、特にどのようにして、あるいは装置が特別な入力促進または可能な状態表示(機構)を持つかは SCORM 1.3 の規定範囲外である。

最良の実施方法としては、対応するイベントが受け付けられるときだけ、RTS はユーザインタフェース装置を利用可能にするべきである。しかしながら、このことはいつでも可能な訳ではない。例えば、“Continue”ユーザインタフェース装置はアグリゲーションの現在の状態を基に利用可能となるが、“Continue”ナビゲーション要求を処理する間に評価されるシーケンシング規則はその要求を拒否してもよい。これは、ナビゲーション要求によって発生するシーケンシング要求の処理の一環で RTS によりアンロードされるまで、履歴データが現 SCO によって報告されなければ起こりえることである。

シーケンシング制御モードは対応するユーザインタフェースの動作を導くべきである。例えば、RTS は“choose”ナビゲーションユーザインタフェース装置を，“choice”シーケンシング制御モードが可能になっているアクティビティクラスタにのみ、有効にすべきである。同様に、RTS は“previous”および“continue”ナビゲーションユーザインタフェース装置を，“flow”シーケンシング制御モードが可能になっているアクティビティクラスタにのみ、有効にすべきである。もし RTS がシーケンシング制御モードに基づいてユーザインタフェース装置を利用可能または不可能の切り替えを行なわない場合には、シーケンシング制御モードに対して有効なナビゲーションイベントおよび対応するシーケンシング要求だけが、RTS によって処理されるべきである。いずれにしろ、シーケンシング要求の処理中に起きるシーケンシング規則の評価が優先し、その結果、シーケンシング要求が拒否される場合がある。言い換えれば、RTS はクラスタの定義されたシーケンシング制御モードやコンテンツアグリゲーションの現状態に基づき、有効とならないナビゲーションイベントおよび対応するシーケンシング要求を廃棄または無視しなければならない。

SCO はナビゲーションイベントや補助サービスイベント用のユーザインタフェース装置を任意に実装できる。SCORM は、SCO が RTS に対して特別なユーザインタフェース装置が利用可能かどうかを決定するために問い合わせる方法、および SCO が RTS に発生したイベントを通知する方法を定義する。

コンテンツ設計者は、アクティビティごとに、アクティビティに対応するコンテンツオブジェクトが起動されるときに、RTS によってデフォルトで提供されるユーザインタフェース装置を利用（視覚化または利用可能）しないことを示すように選択することができる。この指示のひとつの目的は、余分なユーザインタフェース装置によって学習者が混乱するのを避けることである。その装置とは、RTS により提供されるユーザインタフェースや SCO に現れる余分な“previous”や“continue”ボタンである。この特徴の他の利用法は SCORM 1.3 の規定範囲外である。

8.4.2. 補助サービス用のユーザインタフェース装置の管理

RTS は、RTS のユーザインタフェースを通じて、コンテンツ中に定義された補助サービスイベントのすべてを始動させる機能を提供しなければならない。

最良の実施方法として、RTS は、現在のスコープにある補助サービスだけについてのユーザインタフェース装置を利用可能にし、提示しなければならない。どのようにこれらの装置が実装されるか、特にどのようにして、あるいは装置が特別な入力促進または可能な状態表示機構を持つかは SCORM 1.3 の規定範囲外である。

現時点では、SCO は補助サービスイベント用のユーザインタフェース装置を実装できない。SCORM は、SCO が RTS に対して“continue”、“previous”および“exit”イベント用の特別なユーザインタフェース装置が利用可能かどうかを決定

するために問い合わせる方法，および SCO が RTS に発生したイベントを通知する方法を定義する．

8.4.2.1 プレゼンテーション情報モデル

SCORM バージョン 1.3 アプリケーションプロファイルは，コンテンツ開発者が所定の学習アクティビティのプレゼンテーションの特徴を記述できるようにするプレゼンテーションモデルを定義している．このプレゼンテーションモデルの主導的な使用例は，コンテンツ開発者に，RTS がナビゲーションユーザインタフェース制御を提供すべきかどうかを示すメカニズムをあたえることである．コンテンツ開発者は，RTS がユーザインタフェース制御を提供するかどうかをアクティビティごとに定義する選択権を持つ．プレゼンテーションモデルは将来，学習アクティビティプレゼンテーションの他の特徴を持つ可能性がある．

項番	名称	説明	値空間	データ型	既定値
1	Presentation	プレゼンテーションに関する情報	-	-	
1.1	NavigationInterface	ユーザインタフェース制御に関する特徴	-	-	
1.1.1	HideRTSUI	RTS は学習者に特定のイベントを発生させるユーザインタフェース装置を提供してはならないことを示す	一つ以上の語彙トークン	特別定義のトークンを持つ公開された拡張語彙(次表参照)．	(空)

表 8.4.2.1a: SCORM Version 1.3 のプレゼンテーション情報モデル

トークン	定義	説明
Previous	Previous navigation device	このトークンが指定されると，RTS は，このアクティビティクラスタの子が現アクティビティである場合は，“Previous” ナビゲーション装置を表示してはならない．
Continue	Continue navigation device	このトークンが指定されると，RTS は，このアクティビティクラスタの子が現アクティビティである場合は，“Continue” ナビゲーション装置を表示してはならない．
Exit	Exit navigation device	このトークンが指定されると，RTS は，このアクティビティクラスタの子が現アクティビティである場合は，“Exit SCO” ナビゲーション装置を表示してはならない．
Abandon	Abandon navigation device	このトークンが指定されると，RTS は，このアクティビティクラスタの子が現アクティビティである場合は，“Abandon SCO” ナビゲーション装置を表示してはならない．

表 8.4.2.1b: SCORM Version 1.3 ランタイムユーザインタフェース装置クラスボキャブラリ

8.4.2.2 表示 XML バインディング

表示情報の XML バインディングは，コンテンツパッケージングマニフェストの XML スキーマの拡張として扱われる．<presentation>と呼ばれる新しい要素が定義されている．<presentation>要素は<navigationInterface>と呼ばれる単一の副要素を持っている．<navigationInterface>要素は<lmsNavFlowControl>と呼ばれる単一の副要素を持っている．

8.4.2.3 <adlcp:presentation> 要素



説明: <adlcp:presentation> 要素は該当の学習アクティビティに対するプレゼンテーション情報をカプセル化するコンテナ要素である。

繰り返し: <adlcp:presentation> 要素は <item> 要素内に 0 または 1 回現れてよい。

属性:

- なし

要素:

- navigationInterface

8.4.2.3.1 <adlcp:navigationInterface> 要素

説明: <adlcp:navigationInterface> 要素は該当のアクティビティに対するナビゲーションインタフェースのプレゼンテーション要求をカプセル化するコンテナ要素である。

繰り返し: <adlcp:navigationInterface> 要素は<adlcp:presentation> 要素内に 0 または 1 回現れてよい。

属性:

- なし

要素:

- lmsNavFlowControl

8.4.2.3.1.1 <adlcp:hideRTSUI> 要素

説明: <adlcp:hideRTSUI> 要素は、学習者に特定のイベントを始動させるユーザインタフェース装置を RTS が提供してはならないことを示す。この要素がとる値は以下に限定するボキャブラリのうちのひとつである。

- “previous” – これが指定された場合、このアクティビティクラスタの子が現アクティビティであれば、RTS は、“Previous” ナビゲーション装置を表示してはならない。
- “continue” – これが指定された場合、このアクティビティクラスタの子が現アクティビティであれば、RTS は、“Continue” ナビゲーション装置を表示してはならない。
- “exit” – これが指定された場合、このアクティビティクラスタの子が現アクティビティであれば、RTS は、“Exit” ナビゲーション装置を表示してはならない。

- “abandon” – これが指定された場合，このアクティビティクラスタの子が現アクティビティであれば，RTS は，“Abandon” ナビゲーション装置を表示してはならない．

繰り返し: <adlcp:hideRTSUI> 要素は<adlcp:navigationInterface> 要素内に 0 または 1 回以上現れてよい．

属性:

- なし

要素:

- なし

8.4.3. ユーザインタフェース状態またはイベントの交信

8.4.3.1 SCO によるユーザインタフェース装置の状態問合せ

SCO がその中にユーザインタフェース装置を有する場合，RTS がその装置が利用可能か否かを決定するのに基づき，SCO はその装置を利用可とすべきか否かを知ることが望ましい．例えば，コンテンツ作者は，順番として次の SCO が存在する場合にのみ，“Continue” や “Next” ボタンを表示するように SCO を開発する方法を選ぶだろう．SCO はそれ自体では決定権はなく，RTS がそれを知り，SCO はその問合せができる必要がある．同様に，SCO はどのシーケンシング制御モードで動作しているかを知ることがあり，これによって SCO は条件付でユーザインタフェースナビゲーション装置を表示できる．

8.4.3.2 SCORM のユーザインタフェースイベント交信データモデル

SCO は，学習者がナビゲーションイベントを発生させるインタフェース装置をその中に持ったり，持たなかったりする．SCO がそういったユーザインタフェース装置を提供しようがしまいが，SCO は RTS との間でナビゲーションイベントを交信できる．例えば，SCO は LMS との間で，“Previous”，“Continue” および/または “Exit” といったイベントを交信できる．

交信は SCORM API を使用して行われる．しかしながら，この交信で使用される情報モデルは履歴情報や学習者情報の更新に使用される CMI データモデルとは関係がない．ユーザインタフェース装置やイベントについて，SCO と RTS 間で行われる交信にはナビゲーションデータモデルを使用する．

項番	名称	説明	値空間	データ型
1	Navigation	ナビゲーションに関する情報	-	-
1.1	Event	ナビゲーションイベントに関する情報で，SCO は LMS に合図を送ろうとする．	“” “previous” “continue” “exit”	ボキャブラリ (制限付)

1.2	ControlModeEnabled	特定の制御モードが利用可か不可かを示す情報。	-	-
1.2.1	Choice	この要素は、Choice シーケンシング制御モードが SCO を持つクラスタに対し利用可能かどうかを決定するのに使用される。	“true”/“false”	論理型
1.2.2	Flow	この要素は、Flow シーケンシング制御モードが SCO を持つクラスタに対し利用可能かどうかを決定するのに使用される。	“true”/“false”	論理型
1.3	EventPermitted	特定のナビゲーションイベントが許されるかどうかを示す情報。	-	-
1.3.1	Continue	この要素は、Continue ナビゲーションイベントで学習リソースが配信されるか、あるいは Continue ナビゲーションイベントの結果として配信される学習リソースがないかを決定するのに使用される。 注記: この要素は、学習者が Continue ナビゲーションイベントを発生させるユーザインタフェースナビゲーション制御のような機構を SCO が提供すべきかどうかを決定するために使用される。	“true”/“false”	論理型
1.3.2	Previous	この要素は、Previous ナビゲーションイベントで学習リソースが配信されるか、あるいは Previous ナビゲーションイベントの結果として配信される学習リソースがないかを決定するのに使用される。 注記: この要素は、学習者が Previous ナビゲーションイベントを発生させるユーザインタフェースナビゲーション制御のような機構を SCO が提供すべきかどうかを決定するために使用される。	“true”/“false”	論理型

表 8.4.3.2a: ナビゲーション通信データモデル I

8.4.3.3 イベント制限ボキャブラリ

イベントデータモデル要素を構成する値は限られたセットとみなされる。データモデルで定義された値のみが使用できる。以下の表 (表 8.4.3.3a) はその値と意図される動作を示す。

ナビゲーションイベントボキャブラリ	動作
“”	SCO が LMSFinish の呼出しにより通信セッションの終了を示した場合、RTS はどのようなナビゲーションイベントも発生させない。 用法: 既定値
“previous”	SCO が LMSFinish の呼出しにより通信セッションの終了を示した場合、RTS は Previous ナビゲーションイベントを発生させる。 用法: api.LMSSetValue(“nav.event”, “previous”)
“continue”	SCO が LMSFinish の呼出しにより通信セッションの終了を示した場合

	合, RTS は <i>Continue</i> ナビゲーションイベントを発生させる。 使用法: api.LMSSetValue(“nav.event”, “continue”)
“exit”	SCO が LMSFinish の呼出しにより通信セッションの終了を示した場合, RTS は <i>Unqualified Exit</i> ナビゲーションイベントを発生させる。 使用法: api.LMSSetValue(“nav.event”, “exit”)

表8.4.3.3a: イベントデータモデル要素固定ボキャブラリ

他のナビゲーションイベントは含まれず, 現時点では SCO からの通信はない。LMS はすべての定義されたナビゲーションイベントを始動させる機能を提供しなければならない。

8.4.3.4 SCO との通信によるユーザインタフェースイベントごとの RTS 動作

学習者が RTS のユーザインタフェースでナビゲーション要求を発生させると, RTS は直ちに動作する。しかしながら, SCO が RTS とナビゲーション要求イベントを通信する場合, RTS はこの要求をバッファに置き, SCO が LMSFinish を呼び出すことによって現通信セッションを終了させるまで動作しない。言い換えれば, SCO による通信イベントの発生は, 2つの段階が必要である。まず, API 機能の LMSSetValue を使用して必要なイベントを RTS と通信し, その次の段階で, LMSFinish を呼んでイベントを発生させる。

注記: この2段階方式は SCO と RTS が通常の方法で SCO の通信セッションを終了することを可能とする。また LMSFinish はパラメタを許さないが, SCO が (イベント) 要求を通信することを可能とする。

注記: LMSFinish を呼び出すまで, SCO はナビゲーション要求イベントをいくつでも設定することができるが, 以前のナビゲーション要求は新しい要求が設定されるとどれも棄てられる。RTS が SCO からの LMSFinish 処理を終了した後, 最後に記録されたナビゲーション要求のみが動作する。

8.4.3.5 ナビゲーションデータモデルバインディング

ナビゲーション通信データモデルは SCORM ランタイム環境通信データモデルからの拡張である。これは CMI データモデルの一部とは見なされず, したがってプレフィックスには“cmi”が付けられていない。

ドット記法バインディング	詳細
nav.event	SCO が LMS に送信したいナビゲーションイベントに関する情報。 LMS の動作: <ul style="list-style-type: none"> • 初期化: LMS はこの値を初期化する責任はない。このデータ要素は SCO がナビゲーションイベントを指示するのに使用される。 • 準拠性(Conformance): この要素は必須であり LMS によって書き込み専用として実装されなければならない。 SCO の動作: <ul style="list-style-type: none"> • 準拠性: この要素は書き込み専用である。このデータモデル要素については, SCO は LMSGetValue() 要求を呼び出すことが許されない。

	<p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は API エラーコードを “404” に設定しなければならない- 要素は書き込み専用であり, “false”を返す. • LMSSetValue(): SCO が LMS とナビゲーションイベントを交信する時, LMS はナビゲーションイベントを即座には処理しない. ナビゲーションイベントは SCO 交信が終了した後 (LMSFinish() 呼出し) に処理される. <p>API 呼出しの例:</p> <ul style="list-style-type: none"> • LMSSetValue(“nav.event”, “continue”)
nav.control_mode_enabled.choice	<p>この要素は, Choice シーケンシング制御モードが, 要求を出す SCO を含むクラスタで利用可能かどうかを決定するために使用される.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は親クラスタに定義されたシーケンシング規則に基づいてこの値を初期化する責任がある. もし親学習アクティビティが Choice モードを可能にしていれば, LMS はこの値を “true”に, そうでなければ“false”に初期化しなければならない. • 準拠性: この要素は必須であり, LMS によって読み出し専用として実装されなければならない. <p>SCO の動作 Behavior:</p> <ul style="list-style-type: none"> • 準拠性: この要素は読み出し専用である. このデータモデル要素については, SCO は LMSSetValue() 要求を呼び出すことが許されない. <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は, 要求を出す SCO を持つクラスタと関連する制御モードに基づいて関連するブーリアン値を返さなければならない. • LMSSetValue(): LMS は API エラーコードを “403” に設定しなければならない- 要素は読み出し専用であり, “false”を返す. LMS は要求に基づいて要素の状態を変更してはならない. <p>API 呼出しの例:</p> <ul style="list-style-type: none"> • LMSGetValue(“nav.control_mode_enabled.choice”)
nav.control_mode_enabled.flow	<p>この要素は, Flow シーケンシング制御モードが, 要求を出す SCO を含むクラスタで利用可能かどうかを決定するために使用される.</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: LMS は親クラスタに定義されたシーケンシング規則に基づいてこの値を初期化する責任がある. もし親学習アクティビティが flow モードを可能にしていれば, LMS はこの値を “true”に, そうでなければ“false”に初期化しなければならない. • 準拠性: この要素は必須であり, LMS によって読み出し専用として実装されなければならない. <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素は読み出し専用である. このデータモデル要素については, SCO は LMSSetValue() 要求を呼び出すことが許されない. <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は, 要求を出す SCO を持つクラスタと関連する制御モードに基づいて関連するブーリアン値を返さなければならない. • LMSSetValue(): LMS は API エラーコードを “403” に設定しなければならない- 要素は読み出し専用であり, “false”を

	<p>返す。LMS は要求に基づいて要素の状態を変更してはならない。</p> <p>API 呼出しの例:</p> <ul style="list-style-type: none"> • LMSGetValue("nav.control_mode_enabled.flow")
nav.event_permitted.continue	<p>この要素は、Continue ナビゲーションイベントが学習リソースの配信を行うか、あるいは Continue ナビゲーションイベントの結果として学習リソースが何も配信されないかを決定するために使用される。</p> <p>注記：この要素は、学習者が Continue ナビゲーションイベントを発生させるユーザインタフェース制御のような機構を、SCO が提供するかどうかを決定するために使用される。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: 概念的 "activity tree" の状態によって、初期化値は決定されない。 • 準拠性: この要素は必須であり、LMS によって読み出し専用として実装されなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素は読み出し専用である。このデータモデル要素については、SCO は LMSSetValue() 要求を呼び出すことが許されない。 <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMSGetValue() 要求時に、LMS は、現在配信されている SCO、SCORM で定義された適用されるシーケンシング動作、モード、規則、条件にもとづいて、Continue ナビゲーションイベントによって、LMS が "continue" 学習リソースを起動することができるかどうかを決定しなくてはならない場合である。シーケンシング動作は時間制限のある条件に基づいているので、以下のことに注意すべきである。すなわち、この要素に対して SCO が LMSGetValue() 要求をするごとに、LMS はこの要素の戻り値を再評価することが必要である。 <p>LMSSetValue(): LMS は API エラーコードを "403" に設定しなければならない。要素は読み出し専用であり、"false" を返す。LMS は要求に基づいて要素の状態を変更してはならない。</p> <p>API 呼出しの例:</p> <ul style="list-style-type: none"> • LMSGetValue("nav.event_permitted.continue")
nav.event_permitted.previous	<p>この要素は、Previous ナビゲーションイベントが学習リソースの配信を行うか、あるいは Previous ナビゲーションイベントの結果として学習リソースが何も配信されないかを決定するために使用される。</p> <p>注記：この要素は、学習者が Previous ナビゲーションイベントを発生させるユーザインタフェース制御のような機構を、SCO が提供するかどうかを決定するために使用される。</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> • 初期化: 概念的 "activity tree" の状態によって、初期化値は決定されない。 • 準拠性: この要素は必須であり、LMS によって読み出し専用として実装されなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素は読み出し専用である。このデータモデル要素については、SCO は LMSSetValue() 要求を呼び出すことが許されない。 <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMSGetValue() 要求時に、LMS は、現在

	<p>配信されている SCO , SCORM で定義された適用されるシーケンシング動作, モード, 規則, 条件にもとづいて, Previous ナビゲーションイベントによって, LMS が “previous” 学習リソースを起動することができるかどうかを決定しなくてはならない場合である. シーケンシング動作は時間制限のある条件に基づいているので, 以下のことに注意すべきである. すなわち, この要素に対して SCO が LMSGetValue() 要求をすることにより, LMS はこの要素の戻り値を再評価することが必要である.</p> <ul style="list-style-type: none"> • LMSSetValue(): LMS は API エラーコードを “403” に設定しなければならない-要素は読み出し専用であり, “false” を返す. LMS は要求に基づいて要素の状態を変更してはならない. <p>API 呼出しの例:</p> <ul style="list-style-type: none"> • LMSGetValue(“nav.event_permitted.previous”)
--	---

表8.1.4.3a: ナビゲーション送信データモデルのドット記法バインディング

SCORM ナビゲーションデータモデルは SCO を試行している時だけ信頼できる. 例えば, 以下のシナリオ (試行 1) を仮定する:

- SCO は nav.event 要素を “continue” に設定する
- LMSFinish() は SCO と送信 API アダプタとの間の送信終了の結果, 呼び出される.

上記の結果としては, “Continue”ナビゲーション要求が出されるということである.

学習経験をしている期間での後の段階で, ユーザ (学習者) は再度同じアクティビティを提供される (試行 2):

- この試行の間, SCO は nav.event 要素を設定しない
- LMSFinish() は SCO と API アダプタとの間の送信終了の結果, 呼び出される

この場合, “ nav.event ” データモデル要素は試行 1 からは持ち越されず, 以前に設定された “continue” 値も保持しない. この場合, LMSFinish() はナビゲーション要求を始動させず, RTS はアクティビティが終了した後, (次の) イベントを待つ.

8.4.4. ナビゲーションによる学習リソースの終了

8.4.4.1 SCO の終了

LMSFinish() API 機能は SCO が LMS と LMSFinish() API の送信を完了したことを示す. ひとたび LMSFinish() 要求が処理されると, LMS はそれが何であれ, SCO からの最後のナビゲーション要求を処理しなければならない. LMSFinish() の正常な呼出しで示される LMS との送信終了に先立って, “nav.event” 要素をとおして LMS と送信したナビゲーションイベントは効力を持たない. ナビゲーション

ンイベントは SCO が終了するまで、LMS によってバッファリングされる。LMSFinish()呼出しの成功(の場合)に加え、その SCO が正常な LMSFinish()呼出しを実行するのに先立ち、SCO が強制終了させられれば、SCO の試行は終了したとみなされる。

SCO (A)が起動されており、次に LMSInitialize()を呼び出すシナリオを考えてみよう。SCO A の走行中に学習者は LMS から提供されているナビゲーションユーザインタフェース制御を使って他のアクティビティを選択する。それに対して、対応する SCO つまり SCO B が同じブラウザウィンドウに起動され、その結果、前の SCO つまり SCO A は強制的にアンロードされる。この場合、SCO A はアンロードが強制的に行われたことを探知したら LMSFinish()を呼び出さなければならない。RTS の実装は、たとえ SCO A が LMSFinish()呼出しに失敗したとしても、SCO A の交信セッションを終了させなければならない。

SCO A は、LMS との必要な交信および LMSFinish()呼出しを行うアンロードイベントのためのハンドラを実装しなければならない。SCORM は将来的には、SCO の強制終了に先立ち、LMS が SCO に通知する交信機構を導入するであろう。

RTS から発生させられたナビゲーション要求イベントは、常に SCO と交信するナビゲーション要求イベントのすべてに優先する。たとえば、上記のシナリオで SCO A が終了に先立ち RTS とナビゲーション要求イベントの交信を行うことが可能な間は、RTS はそのユーザインタフェースで学習者の操作により他のナビゲーション要求が発生させられても、直ちにそのナビゲーション要求を無視する。上記のシナリオでは、SCO A が RTS と“continue”ナビゲーションイベントを交信し、次に“choose”ナビゲーションイベントにより強制終了させられても、“choose”イベントは RTS から発生させられているので、常に優先する。

9 章

ランタイム環境データモデル

DRAFT

このページは空白である。

DRAFT

9.1.1. 序説

SCORM 内への IMS シンプルシーケンシング仕様の導入に伴い，SCORM のランタイム環境データモデルへの様々な変更が必要となった．今回導入された変更は，P1484.11.1 Data Model for Content to Learning Management System Communication に採用するよう IEEE CMI ワーキンググループへも提案されている．ひとたび IEEE CMI ドラフト標準が承認されると，SCORM にはさらなる変更がもたらせられよう．次節では IMS SS 仕様導入のサポートに必要な変更，これらの変更に基づく幾つかの廃止要素，および，7.1.1 節で議論されたトラッキング状態モデルにどの要素が影響を与えるかを概説する．

9.1.2. SCORM ランタイム環境データモデル

本節では，SCORM ランタイム環境データモデルへの変更点について概説する．変更の幾つかは IMS シンプルシーケンシング仕様を取り入れるためのものであり，他はデータモデルの不明瞭な点を明確化するためのものである．

SCORM ランタイム環境データモデルのあるものとトラッキングモデルおよびアクティビティ状態モデルの対応するアクティビティとの間には幾つかの重要な関係がある．以下の節では，種々の SCORM ランタイム環境データモデル要素とトラッキングモデルまたはアクティビティ状態モデルに与える影響との関係を詳細に述べる．これらの関係は，コンテンツ開発者によりシーケンシング規則で定義された条件で値を設定する影響のために理解しておくことが重要である．

相互運用性を高めるために，SCORM V1.3 ではすべての SCORM ランタイム環境データモデル要素を必須とする．オプションデータモデル要素の概念は LMS の将来展望の観点から削除された．ある要素がこの節に含まれていない場合，それは現在の定義および動作が SCORM V1.2 と変更がない（すべての要素が必須という点を除き）ということを示している．

9.1.2.1 cmi.core.lesson_status

cmi.core.lesson_status 要素は過去に二元的な意味を持っていた．習得状態（合格，不合格）と完了状態（完了，未完了，未試行，閲覧済み）の二つである．シンプルシーケンシングでは，この二つの意味の間に区別を求めており，結果として，cmi.core.lesson_status 要素はなくなり，次の二つの新要素と置き換えられる．

- cmi.core.success_status
- cmi.core.completion_status

9.1.2.1.1 cmi.core.success_status

cmi.core.success_status データモデル要素は、SCO が習得もしくは成功を示すために用いる状態表示である。

cmi.core.success_status	<p>定義: 学習者が関連する SCO を習得したかどうか(合格、不合格)を示す。</p> <p>データ要素詳細:</p> <ul style="list-style-type: none"> • データ型: 語彙 • 値空間: 次の決まった語彙のいずれかでなければならない。 <ul style="list-style-type: none"> ○ 未定(“unknown”) – 学習者が SCO に合格か不合格かを SCO または LMS が断定できないまたは断定していない。 ○ 合格(“passed”) – SCO の学習目標に合格とみなされる。 ○ 不合格(“failed”) – SCO の学習目標に不合格とみなされる。 <p>LMS の動作:</p> <ul style="list-style-type: none"> • 使用法: 成功状態(success status)は様々なシナリオで種々決定される。以下のガイドラインは成功状態を決定するために固守されなければならない。 <ul style="list-style-type: none"> ○ SCO が成功状態 (cmi.core.success_status)を通知し、合格点数が定義されていない場合 (imsss:minNormalizedMeasure), LMS は cmi.core.success_status を記録するために SCO が設定した値を使用しなければならない。 ○ 合格情報 (imsss:minNormalizedMeasure) が提示されており、SCO も正規化得点(cmi.core.score.normalized)を通知する場合、LMS は、正規化得点を合格得点と比較して合格状態を決定しなければならない。SCO も合格状態(success_status)を通知した場合、LMS はこの状態(status)を上書きする。 ○ 合格情報が提示されていない場合、LMS は SCO が設定する状態に依存しなければならない。SCO が状態を通知しない場合、LMS は成功状態を未定(“unknown”)に設定しなければならない。 • 初期化: LMS は success_status を未定(“unknown”)に初期化する責任がある。 • 準拠性: この要素は必須であり、LMS は読み出し/書き込み可能として実装しなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素は SCO からは任意(選択的)に使用される。SCO は必要に応じてこのデータモデル要素を読み書きすることができる。 <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は保持している学習者の関連する success_status を返さなければならない。 • LMSSetValue(): LMS は success_status データモデル要素に SCO から渡された値を設定しなければならない。SCO が success_status を設定するリクエストを呼び、その値が正しいデータ型でない場合、LMS は API エラーコードに 405 - 不正データ型 - を設定し、偽(“false”)を返さなければならない。LMS は不正なリクエストに基づいて要素の状態を変更してはならない。 <p>使用例: LMSGetValue(“cmi.core.success_status”)</p>
-------------------------	--

表9.1.2.1.1a: cmi.core.success_status データモデル要素要求

トラッキングモデル関連: cmi.core.success_status データモデル要素はトラッキングモデルで定義される *Objective Satisfied Status* 要素に対応付けられなくてはならない。トラッキングモデルの性質により *Objective Progress Status* 状態要素も影響を受ける。cmi.core.success_status 要素はロールアップに貢献するアクティビティの主学習目標 (the primary objective) に直接影響を与える。cmi.core.success_status から *Objective Satisfied Status* への対応付けは以下の表に記述されている。

cmi.core.success_status	Objective Satisfied Status
合格("passed")	真(True)
不合格("failed")	偽(False)
未定("unknown")	偽(False)

表9.1.2.1.1b: 成功状態と Objective Satisfied Status の対応

9.1.2.1.2 cmi.core.completion_status

cmi.core.completion_status データモデル要素は SCO が完了を示すために使用される状態表示である。

cmi.core.completion_status	<p>定義: 学習者が関連する SCO の学習を完了したかどうかを示す。</p> <p>データ要素詳細:</p> <ul style="list-style-type: none"> • データ型: 語彙 • 値空間: 次の決まった語彙のいずれかでなければならない。 <ul style="list-style-type: none"> ○ 未定("unknown") –SCO または LMS が他の三つの状態のいずれかと断定できないまたは断定していない。 ○ 完了("completed") – 学習者は、コンテンツ開発者が完了とみなすに十分な程度 SCO を学習した。 ○ 未完了("incomplete") – SCO の学習は開始されたが完了しなかった。 ○ 未試行("not attempted") – 学習者は SCO の学習に取りかかっていない、あるいは学習者は以前 SCO の学習に取りかかったものの、コンテンツ開発者が未試行とみなす程度の少しの学習であった。 <p>LMS の動作:</p> <ul style="list-style-type: none"> • 用法: SCO はその完了状態(completion_status)を LMS に通知しなければならない。LMS は完了状態を決定することができる機構を持たない。 • 初期化: LMS は completion_status を未定("unknown")に初期化する責任がある。 • 準拠性: この要素は必須であり、LMS は読み出し/書き込み可能として実装しなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素は SCO からは任意(選択的)に使用される。SCO は必要に応じてこのデータモデル要素を読み書きすることができる。 <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は現在保持している学習者の関連する完了状態(completion_status)を返さなければならない。 • LMSSetValue(): LMS は完了状態(completion_status)データモデル要素に SCO から渡された値を設定しなくてはならない。SCO が完了状態を設定するために要求呼出しを行い、その値が正しいデータ型でない場合、LMS は API エラーコードに 405 - 不正データ型 - を設定し、偽("false")を返さなければならない。LMS は不正なリクエストに基づいて要素の状態を変更してはならない。 <p>使用例: LMSGetValue("cmi.core.completion_status")</p>
----------------------------	---

表9.1.2.1.2a: cmi.core.completion_status データモデル要素要求

トラッキングモデル関連: cmi.core.completion_status データモデル要素はトラッキングモデルで定義される Attempt Completion Status 要素に対応付けられなくてはならない。トラッキングモデルの性質により、Attempt Progress Status 要素も影響を受ける。cmi.core.completion_status から Attempt Completion Status への対応付けは以下の表に記述されている。

cmi.core.completion_status	Attempt Completion Status
未試行("not attempted")	偽(False)
合格("passed")	真(True)
不合格("failed")	偽(False)
未定("unknown")	偽(False)

表9.1.2.1.2b: 完了状態と試行完了状態の対応

9.1.2.2 cmi.core.entry

SCORM ランタイムデータモデルの cmi.core.entry は、学習者が以前 SCO に入っていた（学習していた）かどうかを示す。

cmi.core.entry	<p>定義: 学習者が以前 SCO に入ったことがあるかどうかの表示。</p> <p>データ要素詳細:</p> <ul style="list-style-type: none"> • データ型: 語彙 • 値空間: 以下の固定語彙のいずれかでなければならない。 <ul style="list-style-type: none"> ○ "ab-initio": これは学習者が最初に SCO に入ることを示す。学習者がプレテストを完了して、SCO の学習目標のすべてに合格したのかもしれないので、未定("unknown") という完了状態(completion_status)は、信頼できる指標とはならない。すなわち、SCO を参照するアクティビティは、学習者が SCO を参照したことがなくても合格となるかもしれない。 ○ "resume"(再入): これは、学習者が以前に SCO に入ったことを示す。学習者は、中断していた SCO を再開している。 ○ ""(空文字列): 空文字列は、初回試行(ab-initio)でも、中断した状態(resume)からの継続でもなく、SCO に入ったことを表わすために使われなければならない。これが使われるシナリオは、SCO がすでに完了した後で、それが復習のために行われた場合である。この場合は、初回試行(ab-initio)でも中断状態からの再開(resume)でもない。 <p>LMS の動作:</p> <ul style="list-style-type: none"> ▪ 使用法: 学習者が最初に SCO に入った時、cmi.core.entry 要素は LMS によって初回試行(ab-initio)に設定されなければならない。学習者が中断した SCO に再入した場合、入り口フラグは LMS によって再開(resume)に設定されなければならない。空文字列は他のすべての場合にも値として使用されなければならない。 ▪ 初期化: LMS は上記の使用法要求を固守しなければならない。 <ul style="list-style-type: none"> ○ 追加動作: LMS が LMSFinish()を受け取るか、または利用者が SCO から離れた場合、LMS は cmi.core.entry に "" (空文字列)または "resume" を設定しなければならない。LMS は、SCO が cmi.core.exit に設定した値を調べることによってこれを決定する。SCO が cmi.core.exit の値に "suspend" を設定すれば、LMS は、次の SCO 起動に備えて、cmi.core.entry に "resume" を設定する。もし、SCO が cmi.core.exit の値に "resume" 以外を設定するかまったく値を設定しない場合、LMS は cmi.core.entry に "" (空文字列)を設定する。 • 準拠性: この要素は必須であり LMS は読み出し専用として実装しなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素は SCO からは任意(選択的)に使用される。SCO は
----------------	---

	<p>必要であれば、この要素を読みだすことのみ許される。</p> <p>API 要求：</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は現在保持している学習者の関連する <code>cmi.core.entry</code> 値を返さなければならない。 • LMSSetValue(): LMS は API エラーコードを 403(要素は読み出し専用)に設定し、偽("false")を返さなければならない。LMS は <code>LMSSetValue()</code> 要求に基づいて要素の状態を変更してはならない。 <p>使用例: <code>LMSGetValue("cmi.core.entry")</code></p>
--	---

表9.1.2.2a: `cmi.core.entry` データモデル要素要求

9.1.2.3 `cmi.core.exit`

SCORM ランタイムデータモデルの `cmi.core.exit` は、学習者がどのようにしてあるいは何故 SCO から退出したかを示す。

<code>cmi.core.exit</code>	<p>定義: 学習者が SCO をどのようにそしてなぜ退出したかの表示。</p> <p>データ要素詳細:</p> <ul style="list-style-type: none"> • データ型: 語彙 • 値空間: 以下の固定語彙のいずれかでなければならない。 <ul style="list-style-type: none"> “Time-Out”: これは、経過時間が長すぎると SCO が断定したか、もしくは <code>max_time_allowed</code>(許容試行時間)を超過したために、SCO が終了したことを示す。<code>max_time_allowed</code> は、マニフェストの中に見出せる(そのアイテムの <code>msss:attemptExperiencedDurationLimit</code>)。 “Suspend”: これは、学習者(彼または彼女)が退出したポイントに、後で戻る意思があって SCO を退出することを示す。 “logout”: これは、学習者が LMS へ戻ってログアウトするのではなく、学習者が SCO 内部からログアウトしたことを示す。これは、SCO がコントロールを LMS システムへ渡し、さらに、LMS システムは - 適切なデータモデル要素を更新した後 - 自動的にコースから学習者をログアウトさせることを意味する。 “”: 空文字列は正常な退出状態を表わすために使用される。 <p>LMS の動作:</p> <ul style="list-style-type: none"> ▪ 使用法: この要素は SCO が最後に退出した理由を示す。SCO は <code>cmi.core.exit</code> に 4 つの値のうちの 1 つを設定する選択権を持つ。その値に基づき、LMS の動作は以下のようである。 <ul style="list-style-type: none"> ○ SCO が <code>cmi.core.exit</code> に “time-out” を設定した場合には、LMS は次の SCO 起動時に、<code>cmi.core.entry</code> に “ ” (空文字列)を設定しなければならない。 ○ SCO が <code>cmi.core.exit</code> に “suspend” を設定した場合には、LMS は次の SCO 起動時に、<code>cmi.core.entry</code> に “resume” を設定しなければならない。 ○ SCO が <code>cmi.core.exit</code> に “logout” を設定した場合には、LMS は次の SCO 起動時に、<code>cmi.core.entry</code> に “ ” (空文字列)を設定しなければならない。さらに、LMS はユーザが起動したどのナビゲーションイベントをも上書きする”Exit All”ナビゲーション要求を発行しなければならない。 ○ SCO が <code>cmi.core.exit</code> に “ ” (空文字列)を設定した場合には、LMS は次の SCO 起動時に、<code>cmi.core.entry</code> に “ ” (空文字列)を設定しなければならない。 ○ SCO が <code>cmi.core.exit</code> に何の値も設定しない場合には、LMS は次の SCO 起動時に、<code>cmi.core.entry</code> に “ ” (空文字列)を設定しなければならない。
----------------------------	--

	<ul style="list-style-type: none"> ▪ 初期化: 要素は初期化される必要はない。LMSGetValue() (の API 呼び出し) はこの要素上で行われてはいない。要素は SCO によってコントロールされる。 ▪ 準拠性: この要素は必須であり、LMS は書き込み専用として実装しなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> ▪ 準拠性: : この要素は SCO からは任意(選択的)に使用される。SCO は必要であれば、この要素を書き込むことのみ許される。 <p>API 要求:</p> <ul style="list-style-type: none"> ▪ LMSGetValue(): LMS は API エラーコードを 404(要素は書き込み専用)に設定し、偽("false")を返さなければならない。 ▪ LMSSetValue(): LMS は、値が固定語彙のうちの一つであれば、その場合のみ cmi.core.exit に指定された値を設定しなくてはならない。要素が固定語彙の一つでない場合、LMS は API エラーコードに 405 -不正データ型 - を設定し、偽(false)を返さなければならない。 <p>使用例:</p> <pre>LMSSetValue("cmi.core.exit", "time-out")</pre>
--	--

表9.1.2.3a: cmi.core.exit データモデル要素要求

アクティビティ状態情報モデル関連: cmi.core.exit モデル要素は、アクティビティ状態情報モデルに定義された *Activity is Suspended* 要素に対応付けられなければならない。SCO が cmi.core.exit データモデル要素を suspend に設定する場合、LMS は *Activity is Suspended* フラグを True に設定しなければならない。SCO が cmi.core.exit データモデル要素を time-out または logout に設定する場合、LMS は *Activity is Suspended* フラグを False に設定しなければならない。注意: logout についての LMS の追加動作については、上記の表を参照。

9.1.2.4 cmi.core.score.normalized

SCORM Version 1.3 では cmi.core.score.normalized 要素を導入する。この要素は、-1 から 1 の間(1 と-1 を含む)に正規化された得点を表す。

cmi.core.score.normalized	<p>定義: 学習者が SCO に対する学習要求をどの程度満足したかを示す数的表示。この得点は確定され、その結果が正規化されていさえすれば、SCO 設計者にとって意味のある何らかの方法で決定され、計算される。</p> <p>cmi.core.score.normalized は-1 から 1 の間(1 と-1 を含む)の値でなければならない。</p> <p>データモデル詳細:</p> <ul style="list-style-type: none"> ▪ データ型: CMIDecimal(数値)または CMIBlack(空白) ▪ 値空間: -1 から 1 の間(1 と-1 を含む)の実数値または空文。 <p>LMS の動作:</p> <ul style="list-style-type: none"> ▪ 使用法: 学習者が SCO への最初の試行である場合、cmi.core.score.normalized は空文字列("")に設定されなければならない。更に試行がある場合には cmi.core.score.normalized は学習者の直近の試行で記録された評価を反映する。SCO に cmi.core.score.normalized が何も設定されておらず、SCO から cmi.core.score.normalized の要求があった場合には、空文字列("")が返されなければならない。 ▪ 初期化: LMS は SCO の最初の起動時に値を空文字列("")に設定しなければならない。SCO はこの値を設定する責任がある。SCO がこの値を設定する前に LMSGetValue() が要求された場合、LMS は空文字列("")を返さなければならない。
----------------------------------	---

	<ul style="list-style-type: none"> ▪ 準拠性：この要素は必須であり，LMS は読み書き可能として実装しなければならない． <p>SCO の動作：</p> <ul style="list-style-type: none"> ▪ 準拠性：この要素は SCO から任意(選択的)に使用される．SCO は必要であれば，この要素に読み書きすることが許される．SCO が正規化点数を保存する必要がある場合は，SCO は-1 から 1 の間(1 と-1 を含む)の値として点数を報告しなければならない． <p>API 要求：</p> <ul style="list-style-type: none"> ▪ LMSGetValue():()：LMS は現在保持している学習者の関連する正規化点数を返さなければならない． ▪ LMSSetValue(): LMS は値が-1 から 1 の間(1 と-1 を含む)の実数表現であれば，cmi.core.score.normalized にその値を設定しなければならない．値が正しいデータ型と範囲値であれば，LMS はエラーコード 0 エラーなし を設定し，真(true)を返さなければならない．要素が正しいデータ型または範囲値でない場合，LMS は API エラーコードに 405 不正データ型 を設定し，偽(false)を返さなければならない． <p>使用例：</p> <ul style="list-style-type: none"> ▪ LMSGetValue("cmi.core.score.normalized"); ▪ LMSSetValue("cmi.core.score.normalized", "0.774");
--	---

表9.1.2.4a: cmi.core.score.normalized データモデル要素要求

トラッキングモデル関連: cmi.core.score.normalized データモデル要素はトラッキングモデルに定義された *Objective Normalized Measure* 要素に対応付けられなければならない．トラッキングモデルの性質上，*Objective Measure Status* も影響を受ける．cmi.core.score.normalized 要素はロールアップに寄与する特定のアクティビティ学習目標の習得度に直接影響を与える．cmi.core.score.normalized から *Objective Normalized Measure* および *Objective Measure Status* への対応付けは次表に記述されている．

cmi.core.score.normalized Value	Objective Measure Status	Objective Normalized Measure
空文字列("")	偽(False)	偽(False)
-1 と 1 の間の得点	真(True)	-1 と 1 の間の得点

表9.1.2.4b: Normalize Score から Objective Normalized Measure への対応付け

9.1.2.5 Raw (cmi.core.score.raw)

SCORM ランタイム環境データモデル要素の *cmi.core.score.raw* は，SCO に対する学習者の最終試行での学習の程度を示す．IMS シンプルシーケンシング仕様の導入により，SCORM はこのデータモデル要素に対しての許容値を変更している．

これは SCORM V1.2 からの変更である．SCORM V1.2 ではこの要素に 0 から 100 までの正規化された値を要求した．この制限は cmi.core.score.normalized 要素の導入により解除される．教材開発者は，今では素点(raw score)をその特定の使用例に適合するどのような方法でも使用できる．

9.1.2.6 Max (cmi.core.score.max)

SCORM ランタイム環境データモデル要素の *cmi.core.score.max* は、学習者が達成可能な最高得点を示す。IMS シンプルシーケンシング仕様の導入により、SCORM はこのデータモデル要素に対しての許容値を変更している。

これは SCORM V1.2 からの変更である。SCORM V1.2 ではこの要素に 0 から 100 までの正規化された値を要求した。この制限は *cmi.core.score.normalized* 要素の導入により解除される。教材開発者は、今では最高得点(max score)をその特定の使用例に適合するどのような方法でも使用できる。

9.1.2.7 Min (cmi.core.min)

SCORM ランタイム環境データモデル要素の *cmi.core.score.min* は、学習者が達成可能な最低得点を示す。IMS シンプルシーケンシング仕様の導入により、SCORM はこのデータモデル要素に対しての許容値を変更している。

これは SCORM V1.2 からの変更である。SCORM V1.2 ではこの要素に 0 から 100 までの正規化された値を要求した。この制限は *cmi.core.score.normalized* 要素の導入により解除される。教材開発者は、今では最低得点(min score)をその特定の使用例に適合するどのような方法でも使用できる。

9.1.2.8 cmi.objectives.n.score.normalized

SCORM V1.3 は *cmi.objectives.n.score.normalized* 要素を導入する。この要素は、-1 から 1 まで(-1, 1 を含む)の間の正規化された得点を示す。

<p>cmi.objectives.n.score.normalized</p>	<p>定義: 学習目標に対する学習者の学習要求満足度の数値的表示である。 <i>cmi.objectives.n.score.normalized</i> は -1 から 1 まで (-1, 1 を含む) の間の正規化された値でなければならない。</p> <p>データモデル詳細:</p> <ul style="list-style-type: none"> • データ型: CMIDecimal (数値) または CMIBlack (空白) • 値空間: 1 から 1 まで (-1, 1 を含む) の間の正規化された値または空文字列 <p>LMS の動作:</p> <ul style="list-style-type: none"> ▪ 使用法: ▪ 初期化: LMS は要素を空文字列("") に初期化しなければならない。SCO はこの値を設定する責任がある。SCO がこの値を設定する前に <i>LMSGetValue()</i> が呼び出された場合、LMS は空文字列("") を返さなければならない。 ▪ 準拠性: この要素は必須であり、LMS は読み書き可能として実装しなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> ▪ 準拠性: LMS は要求を識別する学習目標正規化得点を返さなければならない。返された値は関連するデータ型と一致しなければならない。学習目標の正規化得点が SCO で設定されたものでなければ、LMS は空文字列("") を返さなければならない。 <p>API 要求:</p> <ul style="list-style-type: none"> ▪ LMSGetValue(): LMS は要求により識別される対象正規化得点を返さなければならない。返された値は関連するデータ型と合致しなければならない。 ▪ LMSSetValue(): LMS は値が -1 から 1 の間 (1 と -1 を含む) の実数表
---	--

	<p>現であれば, cmi.core.score.normalized にその値を設定しなければならない. 値が正しいデータ型と範囲値であれば, LMS はエラーコード 0 エラーなし を設定し, 真(true)を返さなければならない. 要素が正しいデータ型または範囲値でない場合, LMS は API エラーコードに 405 不正データ型 を設定し, 偽(false)を返さなければならない.</p> <p>使用例: LMSGetValue("cmi.objectives.3.score.normalized") LMSSetValue("cmi.objectives.0.score.normalized", "0.75") }</p>
--	---

表9.1.2.8a: cmi.objectives.n.score.normalized データ型/要素要求

トラッキングモデル関連:

cmi.objectives.n.score.normalized データモデル要素はトラッキングモデルに定義された *Objective Normalized Measure* 要素に対応付けられなければならない. トラッキングモデルの性質により, *Objective Normalized Measure* 要素もまた影響を受ける. cmi.objectives.n.score.normalized 要素は, cmi.objectives.n.id の識別子に対応するアクティビティの特定な学習目標の習得度に直接影響を与える.

cmi.core.score.normalized から *Objective Normalized Measure* および *Objective Measure Status* への対応付けは以下の表に記述されている:

cmi.objectives.n.score.normalized Value	Objective Measure Status	Objective Normalized Measure
空文字列("")	偽(False)	偽(False)
-1 と 1 の間の得点	真(True)	-1 と 1 の間の得点

表9.1.2.8b: 正規化得点に基づく状態の決定

以下の擬似コードは, 学習目標の正規化得点から *Objective Normalized Measure* および *Objective Measure Status* への対応付けの概念過程を説明している.

```

Loop over the objective array cmi.objectives._count times
  Read the cmi.objectives.<loopIndex>.id.
  If the activity has an objective with a corresponding ID
    Set the associated Objective Measure Status.
    Set the associated Objective Normalized Measure.
  Else
    Ignore the SCO set normalized score.
  End If
End Loop

```

9.1.2.9 cmi.objectives.n.score.raw

SCORM ランタイム環境データモデル要素の *cmi.objectives.n.score.raw* は, 学習目標に対する学習者の最終試行での習得度を示す. IMS SS 仕様の導入により, SCORM はこのデータモデル要素に対しての許容値を変更している. これは SCORM V1.2 からの変更である. SCORM V1.2 ではこの要素に 0 から 100 までの正規化された値を要求した. この制限は cmi.objectives.n.score.normalized 要素の導入により解除される. 教材開発者は, 今では素点(raw score)をその特定の使用例に適合するどのような方法でも使用できる.

9.1.2.10 cmi.objectives.n.score.max

SCORM ランタイム環境データモデル要素の *cmi.objectives.n.score.max* は、学習者が達成可能な最高得点を示す。IMS シンプルシーケンシング仕様の導入により、SCORM はこのデータモデル要素に対しての許容値を変更している。これは SCORM V1.2 からの変更である。SCORM V1.2 ではこの要素に 0 から 100 までの正規化された値を要求した。この制限は *cmi.objectives.n.score.normalized* 要素の導入により解除される。教材開発者は、今では最高得点(max score)をその特定の使用例に適合するどのような方法でも使用できる。

9.1.2.11 *cmi.objectives.n.score.min*

SCORM ランタイム環境データモデル要素の *cmi.objectives.n.score.min* は、学習者が達成可能な最低得点を示す。IMS シンプルシーケンシング仕様の導入により、SCORM はこのデータモデル要素に対しての許容値を変更している。これは SCORM V1.2 からの変更である。SCORM V1.2 ではこの要素に 0 から 100 までの正規化された値を要求した。この制限は *cmi.objectives.n.score.normalized* 要素の導入により解除される。教材開発者は、今では最低得点(min score)をその特定の使用例に適合するどのような方法でも使用できる。

9.1.2.12 *cmi.objectives.n.status*

cmi.objectives.n.status 要素は、過去には二つの意味を持っていた。習得状態（合格、不合格）と完了状態（完了、未完了、未試行、閲覧）である。シンプルシーケンシングはこの二つに明確な区別を要求する。その結果、*cmi.objectives.n.status* 要素は、なくなり、次の二つの新要素と置き換えられる:

- *cmi.objectives.n.success_status*
- *cmi.objectives.n.completion_status*

9.1.2.12.1 *cmi.objectives.n.success_status*

cmi.objectives.n.success_status データモデル要素は、SCO が学習目標の習得状況ないし成功(状態)を示すために使用しなければならない状態表示である。

<i>cmi.objectives.n.success_status</i>	<p>学習者が関連する学習目標を習得しかたどうか(合格、不合格)を示す。</p> <p>データ要素詳細:</p> <ul style="list-style-type: none"> • データ型: 語彙 • 値空間: 以下の固定用語のいずれかでなければならない。 <ul style="list-style-type: none"> ○ 未定(“unknown”) – SCO または LMS は学習者が学習目標に合格か不合格かを断定できないかもしくは断定していない。 ○ 合格(“passed”) – 学習目標に合格とみなす。 ○ 不合格(“failed”) – 学習目標に不合格とみなす。 <p>LMS の動作:</p> <ul style="list-style-type: none"> • 使用法: 通常、SCO は LMS にその学習目標の成功状態を通知する。 <ol style="list-style-type: none"> (1) コンテンツ開発者が習得目標情報を提示している場合 (<i>imsss:minNormalizedMeasure</i>)、LMS は学習者の得点習得 (<i>cmi.objectives.n.score.normalized</i>) を習得必要得点
--	--

	<p>(imsss:minNormalizedMeasure)と比較して、成功状態を合格か不合格に変更しなければならない。</p> <p>(2) コンテンツ開発者が習得目標情報を提示していない場合、LMS は SCO から報告される成功状態(success_status)に依存しなければならない。</p> <ul style="list-style-type: none"> • 初期化: LMS は成功状態(success_status)を未定("unknown")に初期化する責任がある。 • 準拠性: この要素は必須要件であり、LMS は読み書き可能として実装されなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素は SCO の使用は任意である。SCO は必要であれば、このデータモデル要素を読み書きできる。 <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は現在保持している関連する成功状態(success_status)を学習者に返さなければならない。 • LMSSetValue(): LMS は、成功状態(success_status)データモデル要素に、SCO から渡される値を設定しなければならない。SCO が成功状態(success_status)の設定要求呼出しを行い、その値が正しいデータ型でない場合、LMS は API エラーコードを"405" - 不正データ - に設定し、偽 ("false")を返さなければならない。LMS は不正要求に基づいて要素の状態を変更してはならない。 <p>使用例: LMSGetValue("cmi.objectives.0.success_status")</p>
--	---

表9.1.2.12.1a: *cmi.objectives.n.success_status* データモデル要素要求

トラッキングモデル関連: *cmi.objectives.n.success_status* データモデル要素はトラッキングモデルに定義された *Objective Satisfied Status* 要素に対応付けられなければならない。トラッキングモデルの性質により *Objective Progress Status* 要素も影響を受ける。*cmi.objectives.n.success_status* 要素は *cmi.objectives.n.id* と一致する識別子を持つアクティビティの特定学習目標に直接影響を与える。*cmi.objectives.n.success_status* から *Objective Satisfied Status* および *Objective Progress Status* への対応付けは以下の表に記述されている。

<i>cmi.objectives.n.success_status</i> value	Objective Satisfied Status
合格("passed")	真(True)
不合格("failed")	偽(False)
未定("unknown")	偽(False)

表9.1.2.12.1b: *cmi.objectives.n.success_status* から *Objective Satisfied Status* への対応付け

以下の擬似コードは学習目標の正規化得点を適切な *Objective Normalized Measure* および *Objective Measure Status* に対応付ける過程を表している。

```

Loop over the objective array cmi.objectives._count times
  Read the cmi.objectives.<loopIndex>.id.
  If the activity has an objective with a corresponding ID
    Set the associated Objective Progress Status.
    Set the associated Objective Satisfied Status.
  Else
    Ignore the SCO set success status.
  End If
End Loop

```

9.1.2.12.2 *cmi.objectives.n.completion_status*

cmi.objectives.n.completion_status データモデル要素は、SCO が学習目標の完了を示すために使用しなければならない状態表示である。

cmi.objectives.n.completion_status	<p>学習者が関連する学習目標を完了したかどうかを示す。</p> <p>データ要素詳細:</p> <ul style="list-style-type: none"> • データ型: 語彙 • 値空間: 以下の固定用語のいずれかでなければならない。 <ul style="list-style-type: none"> ○ 未定(“unknown”) – SCO または LMS は他の三つの状態と断定できないかまたはその中の一つと断定していない。 ○ 完了(“completed”) – 学習者は、コンテンツ開発者が完了とみなせる程、学習目標を経験した。 ○ 未完了(“incomplete”) – 学習目標(の学習)を開始したが修了していない。 ○ 未試行(“not attempted”) – 学習者はその学習目標にアクセスしていない、もしくは以前にアクセスしたもののほんのわずかに経験しただけなので、コンテンツ開発者は未試行とみなす。 <p>LMS の動作:</p> <ul style="list-style-type: none"> • 使用法: SCO は LMS に対し学習目標の完了状態(completion_status)を通知しなければならない。LMS は完了状態を決定することができる機構を持たない。 • 初期化: LMS は完了状態(completion_status)を未定(“unknown”)に初期化する責任がある。 • 準拠性: この要素は必須要件であり、LMS は読み書き可能として実装しなければならない。 <p>SCO の動作:</p> <ul style="list-style-type: none"> • 準拠性: この要素の使用は SCO には任意である。SCO は必要であれば、このデータモデル要素を読み書きできる。 <p>API 要求:</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS は現在保持している関連する完了状態(completion_status)を学習者に返さなければならない。 • LMSSetValue(): LMS は、完了状態(completion_status)データモデル要素に、SCO から渡される値を設定しなければならない。SCO が完了状態(completion_status)の設定要求呼出しを行い、その値が正しいデータ型でない場合、LMS は API エラーコードを“405” – 不正データ型 - に設定し、偽 (“false”)を返さなければならない。LMS は不正要求に基づいて要素の状態を変更してはならない。 <p>使用例: LMSGetValue(“cmi.objectives.n.completion_status”)</p>
------------------------------------	--

表9.1.2.12.2a: cmi.objectives.n.completion_status データモデル要求

9.1.2.13 cmi.student_data.mastery_score

cmi.student_data.mastery_score 要素は SCO 外で決定される SCO の合格正規化得点を示すものである。

cmi.student_data.mastery_score	<p>定義: SCO 外で決定されたものとして、SCO の学習要求を満足するよう要求される学習者得点。SCO の正規化得点が習得必要得点(mastery_score)以上である場合、学習者は合格した、あるいはその教材を習得したとみなされる。ある場合には、SCO は、この渡された得点が LMS によって決定されるために、どのような点数かを知らない。</p> <p>使用法: 習得必要得点(mastery_score)をサポートする LMS については、渡された得点に基づいて(それが SCO からのものであれば)、成功状態</p>
--------------------------------	---

	<p>(success_status)を変更することができなければならない。習得必要得点 (mastery_score)の SCO への引渡しはこの機能のフルサポートとはならない。</p> <p>形式: -1 から 1 まで (-1, 1 を含む)の間の実数値</p> <p>LMS の動作:</p> <p>初期化: LMS はマニフェストから得られる(マニフェストに定義されていれば)値で要素を初期化する責任がある。初期化には item の初期学習目標である<imsss:minNormalizedMeasure> が使用される。</p> <ul style="list-style-type: none"> ▪ LMSGetValue(): LMS が格納した現在値を返す。 <ul style="list-style-type: none"> ○ 戻り値の例: <ul style="list-style-type: none"> “0.75” “-0.23” “.5” ▪ LMSSetValue(): LMS は以下に基づいてエラーコードを設定しなければならない。 <ul style="list-style-type: none"> ○ エラーコード: <ul style="list-style-type: none"> ▪ 403 – 要素は読み出し専用。 この要素を LMS がサポートしており、この要素に対して LMSSetValue()を起動する要求があった場合、LMS はエラーコードを 403 に設定しなければならない。 <p>SCO 使用例:</p> <pre>var masteryScoreValue = LMSGetValue("cmi.student_data.mastery_score");</pre>
--	--

表9.1.2.13a: cmi.student_data.mastery_score データモデル要求

9.1.2.14 cmi.student_data.max_time_allowed

cmi.student_data.max_time_allowed 要素は、SCO のいかなる試行にも許される最大時間を示すものである。

<p>cmi.student_data.max_time_allowed Read Only</p>	<p>定義: 学習者が現在試行中の SCO に対して許される学習時間の総計。制限時間の超過に対して SCO に期待される応答については time_limit_action を参照。</p> <p>使用法: 学習者が SCO 内で(学習に)許される最大時間を SCO に提示するために使用される。</p> <p>形式: コロンで区切られた時, 分, 秒。HHHH:MM:SS.SS 時は最小 2 けたの数字で 最大 4 けたの数字である。分は 2 けたの数字だけでなければならない。秒は 2 けたの数字と、付加的に少数点に加えて 1 けたまたは 2 けたの数字でなければならない。(例. 34.45).</p> <p>LMS の動作:</p> <ul style="list-style-type: none"> ▪ 初期化: 要素がマニフェスト中にあれば、LMS はマニフェストから得られる値(ActivityExperiencedDurationLimit)で要素を初期化する責任がある。 ▪ LMSGetValue(): LMS が格納した現在の値を返す。 <ul style="list-style-type: none"> ○ 戻り値の例: <ul style="list-style-type: none"> “00:14:30” “02:03:00” “01:09:00” ▪ LMSSetValue(): LMS は以下に従ってエラーコードを設定しなければならない。 <ul style="list-style-type: none"> ○ エラーコード: <ul style="list-style-type: none"> ▪ 403 – 要素は読み出し専用。 この要素を LMS がサポートしており、この要素に対して LMSSetValue()を起動する要求があった場合、LMS はエラーコードを 403 に設定しなければならない。
---	---

	<p style="text-align: center;">ない。</p> <p>SCO 使用例: var maxTimeAllowedValue = LMSGetValue("cmi.student_data.max_time_allowed");</p>
--	---

表9.1.2.14a: *cmi.student_data.max_time_allowed* データモデル要求

DRAFT

このページは空白である。

DRAFT

付録 A 略語表

DRAFT

このページは空白である。

DRAFT

略語表

ADL	Advanced Distributed Learning
API	Application Program Interface
CMI	Computer Managed Instructions
DoD	Department of Defense
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
LMS	Learning Management System
LOM	Learning Objects Metadata
LTSC	Learning Technology Standards Committee
MD	Meta-data
MIME	Multipurpose Internet Mail Extensions
PIF	Package Interchange Format
RTE	Run-Time Environment
RTS	Run-Time Service
SCA	Sharable Content Asset
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SS	Simple Sequencing
URI	Universal Resource Identifier
URL	Universal Resource Locator
XSD	XML Schema Definition
XML	eXtensible Markup Language

このページは空白である。

DRAFT

付録 B 参考文献

DRAFT

このページは空白である。

DRAFT

参考文献

1. Final 1484.12.1-2002 Learning Object Metadata draft standard
Available at: <http://www.ieee.org/>
2. IEEE P1484.12.3/D2, 2003-01-17 Draft Standard for XML Binding for Learning Object Metadata
Available at: *To Be Supplied by IEEE LTSC*
3. IMS Simple Sequencing Specification Version 1.0 Final Release
Includes:
 - IMS SS Information and Behavior Model
 - IMS SS Best Practice and Implementation Guide
 - IMS SS XML BindingAvailable at <http://www.imsglobal.org/>

DRAFT

このページは空白である。

DRAFT

付録 C

改定履歴

DRAFT

このページは空白である。

DRAFT

改定履歴

Version	Comment
Version 0.9 <i>Draft</i>	初期開放ドラフト版
Version 0.9.2 <i>Draft</i>	IMS シンプルシーケンシング仕様の最終版および IEEE P1484.12.3 のドラフト版に基づき更新
Version 1.0 <i>Draft</i>	IMS シンプルシーケンシング仕様の最終版および IEEE P1484.12.3 XML Binding Draft Version 1 に基づき最終更新

DRAFT

このページは空白である。

DRAFT