

平成 18 年度
学士学位論文

**データモデル及び API インスタンスの利
用に着目した SCORM コンテンツ作成支
援システムの構築**

Development of SCORM Making Support System
focus on using Data Model and API Instance

1070419 大黒 隆弘

指導教員 妻鳥 貴彦

2007 年 3 月 9 日

高知工科大学 情報システム工学科

要 旨

データモデル及び API インスタンスの利用に着目した SCORM コンテンツ作成支援システムの構築

大黒 隆弘

近年 e-Learning の標準規格である SCORM (Sharable Content Object Reference Model) が注目されている。SCORM は、e-Learning コンテンツの再利用性や相互運用性の向上を目指している。e-Learning に SCORM を適用すると、コンテンツの再利用性や相互運用性が向上するばかりでなく、SCORM 教材 (SCO) と学習管理システム (LMS) の間でやりとりできるデータが充実することで高度な学習管理を可能にする。しかし SCORM コンテンツの作成にはコンテンツ作成者に対して SCORM の複雑な定義や HTML、XML、JavaScript の記述に関する知識が要求されるため、これらの要因が教材作りに専念することを妨げている。特に、ランタイム環境における SCO (Sharable Content Object) と LMS (Learning Management System) の間の通信を充実させる JavaScript の記述は、その知識を持たないコンテンツ作成者には困難である。

そこで本研究では、SCORM の知識などを持たないコンテンツ作成者でも SCORM コンテンツを作成できるようにするために、SCORM コンテンツ作成支援システムを構築した。本稿では特に、SCO と LMS の間のデータ通信を制御しているデータモデル及び API インスタンスの利用に着目し、問題文や配点等の入力のみで演習やテスト用の SCORM コンテンツを作成できることを示す。また、本システムを用いて作成したコンテンツが、SCORM コンテンツとして機能していることを確認するために、SCORM 対応の LMS に登録して確認作業を行った。その結果、LMS に学習履歴情報が反映されていることが確認できた。

キーワード SCORM、ランタイム環境、SCORM コンテンツ

Abstract

Development of SCORM Making Support System focus on using Data Model and API Instance

Takahiro DAIKOKU

Recently, SCORM (Sharable Content Object Reference Model) has been spreading in Web-Based e-Learning as standard. SCORM aims improving reusability and interoperability in e-Learning. Especially, RTE (Run-Time Environment) provides more effective learning through the communication between SCO (Sharable Content Object) and LMS (Learning Management System). However, SCORM content authors need definite knowledge of SCORM. SCORM content are described in HTML, XML and JavaScript. Therefore, the authors need to know them preliminarily. Describing them interfere the authors to concentrate on making contents. Especially descriptions of the communication between SCO and LMS in JavaScript is difficult.

In this research, we focused on using Data Model and API Instance in RTE. We developed SCORM Making Support System. This system supports authors to describe RTE in SCORM content. And we confirmed content made by this system on SCORM compliant LMS.

key words SCORM, Run-Time Environment, SCORM Content

目次

第 1 章	はじめに	1
第 2 章	背景	2
2.1	SCORM 規格	2
2.2	SCORM2004	3
2.3	コンテンツアグリゲーションモデル	3
2.3.1	アセットと SCO	4
2.3.2	コンテンツの構造化	4
2.3.3	メタデータ	5
2.4	ランタイム環境の概要	5
2.4.1	SCO と LMS の間の通信	5
2.4.2	データ通信の実現	6
2.5	SCORM コンテンツ作成における問題点	6
第 3 章	ランタイム環境	7
3.1	SCORM のデータモデル	7
3.1.1	データ型の種類	8
3.1.2	SPM	9
3.1.3	マニフェストファイルの記述内容で設定するデータモデル要素	9
3.1.4	コレクションとして定義されるデータモデル要素	10
3.2	API インスタンス	11
3.2.1	API_1484_11	12
3.2.2	API 探索アルゴリズム	12
3.2.3	データ転送関数	12

目次

3.2.4	通信制御関数	13
3.2.5	エラー関数	14
第 4 章	SCORM コンテンツ作成支援システムの設計	15
4.1	システム構築の目的	15
4.2	作成対象となる SCORM コンテンツ	15
4.3	支援方法	16
4.4	システムの動作	17
4.5	テンプレートファイル	17
4.5.1	HTML のテンプレートファイル	17
4.5.2	XML のテンプレートファイル	18
4.5.3	JavaScript のテンプレートファイル	19
4.6	SCORM コンテンツを構成するファイル	20
第 5 章	SCORM コンテンツ作成支援システムの構築	22
5.1	システムの概要	22
5.1.1	フォルダ名・ファイル名の入力	23
5.1.2	作成可能な問題	24
	正誤判定問題	24
	単一選択問題	24
	複数選択問題	25
	記述問題	25
	並べ替え問題	26
	組み合わせ問題	26
	穴埋め問題	27
5.1.3	その他の入力事項	27
5.2	動作確認	29

目次

第 6 章 おわりに	31
謝辞	32
参考文献	33
付録 A データモデル概要	34

目次

2.1	コンテンツの構造化	4
2.2	ランタイム環境	5
3.1	ランタイム環境における SCO と LMS の通信メカニズム	7
3.2	レコードタイプのデータモデル	11
3.3	API インスタンスの探索例	12
5.1	コンテンツ作成までの画面遷移	23
5.2	問題作成画面（左）と学習画面（右）	28
5.3	問題作成画面と学習画面（続き）	29
5.4	LMS から起動した SCO の学習画面	30
5.5	学習進捗ログ表示画面（上）と演習回答ログ表示画面（下）	30

表目次

3.1	マニフェストファイルの記述内容で設定するデータモデル要素	10
4.1	インタラクションのデータのタイプ	16
A.1	SCORM2004 のデータモデル要素の概要	34
A.2	SCORM2004 のデータモデル要素の概要 (続き)	35

第 1 章

はじめに

近年、e-Learning コンテンツの相互運用性や再利用性を高めるための標準規格である SCORM が注目されている。SCORM に準拠したコンテンツは、異なる SCORM 対応 LMS (Learning Management System) で共有・利用することが可能である。現在では SCORM2004 が規格の最新版である [1][2]。SCORM では、LMS と SCO (Sharable Content Object) の間でやり取りされる得点や学習の完了状態など、高度な学習管理を実現するための情報も定義されている。その反面、規格の定義は複雑で、コンテンツ作成者には高度な SCORM の知識が要求される。また、SCORM コンテンツ作成には HTML、XML、JavaScript の記述が必要であり、これらの要因は教材作りに専念したいコンテンツ作成者にとってコンテンツ作成の妨げとなっている。

本研究では、特に SCORM2004 のデータモデル及び API インスタンスの利用に着目して、SCORM や HTML、XML、JavaScript に関する知識がなくても演習あるいはテストに用いる SCORM コンテンツを容易に作成できるシステムを構築する。また、SCORM 対応 LMS に登録することでコンテンツの動作確認を行う。

第 2 章

背景

近年、インターネットを活用した学習形態である e-Learning が注目を浴びている。これにより、インターネットに接続できる端末から、いつでもどこでも学習できるようになった。現在、この e-Learning は大学の講義や企業研修などで活用されている。そして、この e-Learning による学習は、主に LMS と呼ばれる学習管理システムを利用することで、各学習者の学習管理を行っている。しかし、e-Learning の学習教材であるコンテンツは、利用できる LMS が限定されていた。例えば、A というコンテンツは A 社の LMS に対応、B というコンテンツは B 社の LMS に対応といったように、LMS が異なれば利用できるコンテンツも異なることが問題であった。

2.1 SCORM 規格

SCORM とは、e-Learning の標準化を目指して策定された標準規格のことである。この規格の策定に至るまでに、様々な団体が e-Learning に関する標準化の検討を行ってきた。それには次のような主要団体がある。

- AICC (Aviation Industry CBT Committee)
- ADL (The Advanced Distributed Learning)
- IMS (The Instructional Management Systems)
- IEEE (The Institute of Electrical and Electronics Engineers)

2.2 SCORM2004

このうち、米国国防総省が中心となって運営している ADL が、他の団体が提唱する標準規格をまとめ上げたことにより、SCORM が e-Learning の標準規格となった。

この規格は、コンテンツの相互運用性や再利用性を向上させるために策定されたものである。具体的に述べると、異なる LMS を用いても同じひとつのコンテンツを利用することや、既にある学習資源を別の SCORM コンテンツに埋め込んで再利用することが可能になった。

SCORM 規格の定義は、コンテンツアグリゲーションモデルとランタイム環境の二つに分けられる。さらに SCORM2004 では、これらに加えシーケンシングとナビゲーションに関しても定義されている。

2.2 SCORM2004

SCORM2004 は、SCORM における最新版の規格である（2007 年現在）。この SCORM2004 の大きな特徴は、新たにシーケンシングとナビゲーションについて定義されたことである。これらの特徴としては各学習者個人の学習進捗状況に応じて学習内容に変化を与えることが可能になったことが挙げられる。

SCORM2004 では、ランタイム環境におけるデータモデルの定義に大きな変更があった。例えば、廃止されたデータモデル要素や新たに増えたデータモデル要素があること、学習者の演習回答や正答情報（インタラクション）の記述に関するフォーマットが詳細化したことなどである。

2.3 コンテンツアグリゲーションモデル

コンテンツアグリゲーションモデルに関する定義では、コンテンツ構造、メタデータ、SCO やアセットについて定義されている。

2.3 コンテンツアグリゲーションモデル

2.3.1 アセットと SCO

アセットとは、最小単位の学習資源ファイルのことである。例えば、画像ファイルのデータや音声ファイルなどがアセットに相当する。このアセットは、LMS と直接通信しないのが特徴である。また、SCO とは、LMS と直接通信を行う学習資源のことであり、HTML により記述される。この SCO に単一あるいは複数のアセットを埋め込むことで利用する場合もある。

2.3.2 コンテンツの構造化

SCORM では、マニフェストファイルと呼ばれる XML 形式のファイルを記述することでコンテンツを構造化することができる。図 2.1 における“organization”とは、SCORM コンテンツの構造体のことであり、“resources”とはその構造体を構成する SCO やアセットのことを指す。

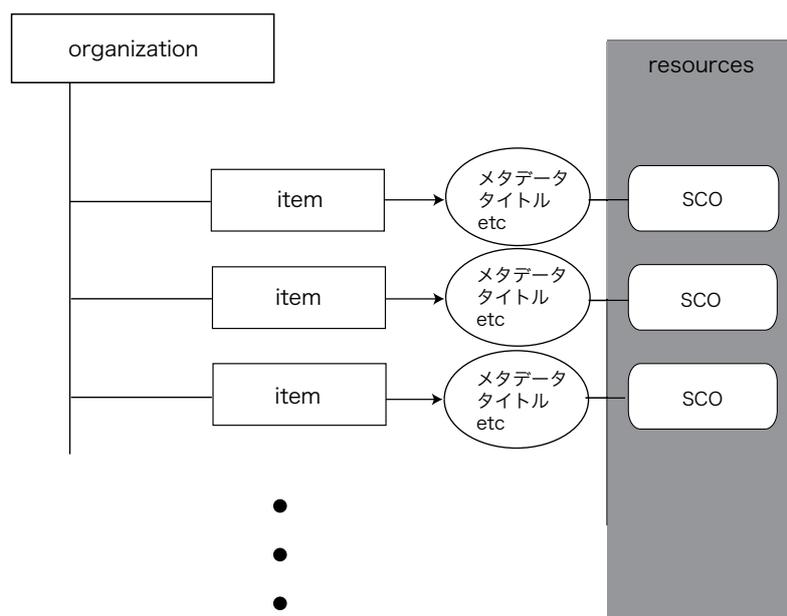


図 2.1 コンテンツの構造化

2.4 ランタイム環境の概要

2.3.3 メタデータ

SCORMにおけるメタデータとは、SCORM コンテンツを構成する SCO やアセットひとつひとつに対して持たせる個別情報のことである。これはコンテンツの再利用や相互運用を可能にするための必要な情報である。メタデータの種類には、著作権やスキーマのバージョン等があり、それらは SCORM の定義で記述が必須であるものとそうでないものに分けられる。SCORM2004 では、< schema >< schemaversion >< metadata > がマニフェストを記述するための必須メタデータとなっている。

2.4 ランタイム環境の概要

ランタイム環境とは、SCORM 教材 (SCO) と、学習管理システム (LMS) の間において行われるデータ通信の実現方法、その際用いられるデータモデルや API インスタンスに関する定義のことである。

2.4.1 SCO と LMS の間の通信

SCO と LMS の間では、合格点や学習者の回答などがデータとしてやり取りされている (図 2.2)。これらのデータは、データモデルとして定義され、API インスタンスを介してやり取りされている。

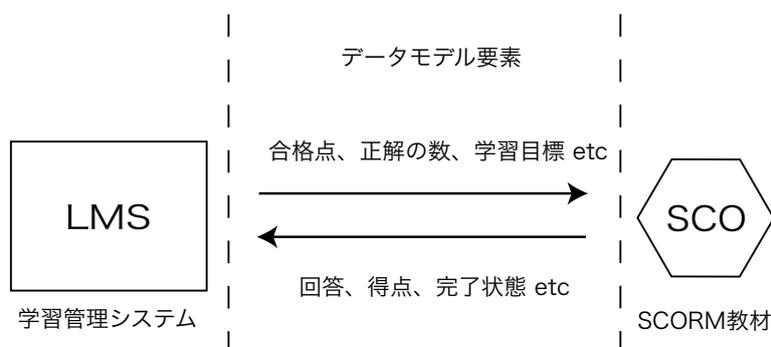


図 2.2 ランタイム環境

2.5 SCORM コンテンツ作成における問題点

2.4.2 データ通信の実現

学習者が、SCO 学習を開始してから終了するまでに行われる処理を記述すると以下のようになる。

1. API インスタンスの探索
2. 通信の初期化
3. 任意データのやり取り
4. 通信の終了

まず SCO は、LMS 側で用意されている API インスタンスの探索を行う。API インスタンスの探索に成功すると直ちに通信の初期化処理が実行され、この処理の直後は学習中の状態となる。学習中はデータモデル及び API インスタンスの利用により SCO と LMS の間で任意のデータをやりとりさせる。そして、学習終了あるいは別の SCO への遷移の際には通信を終了させる必要がある。

2.5 SCORM コンテンツ作成における問題点

SCORM コンテンツ作成における問題点は、SCORM コンテンツ作成を複雑にしている要因が数多く存在することである。これらの要因は、コンテンツ作成者が教材作りに専念することの妨げになっている。また、これらは教材そのものの質を落とすことにもつながる。

SCORM コンテンツ作成を困難にしている要因は、次のようにまとめることができる。

- SCORM コンテンツの作成には、SCORM に関する複雑な知識が必要である。
- JavaScript 記述によるランタイム環境の実現が必要である。
- XML 記述によるコンテンツアグリゲーションモデルの実現が必要である。
- データモデル要素ごとに読み書き制限の違いやデータ型の違いがあるため複雑である。

第3章

ランタイム環境

ランタイム環境では、SCO と LMS の間における通信のメカニズム（図 3.1）及びデータモデルや API インスタンスに関して定義されており、JavaScript 記述によりこの通信メカニズムを実現している。この章では、データモデルと API インスタンスについて述べる。

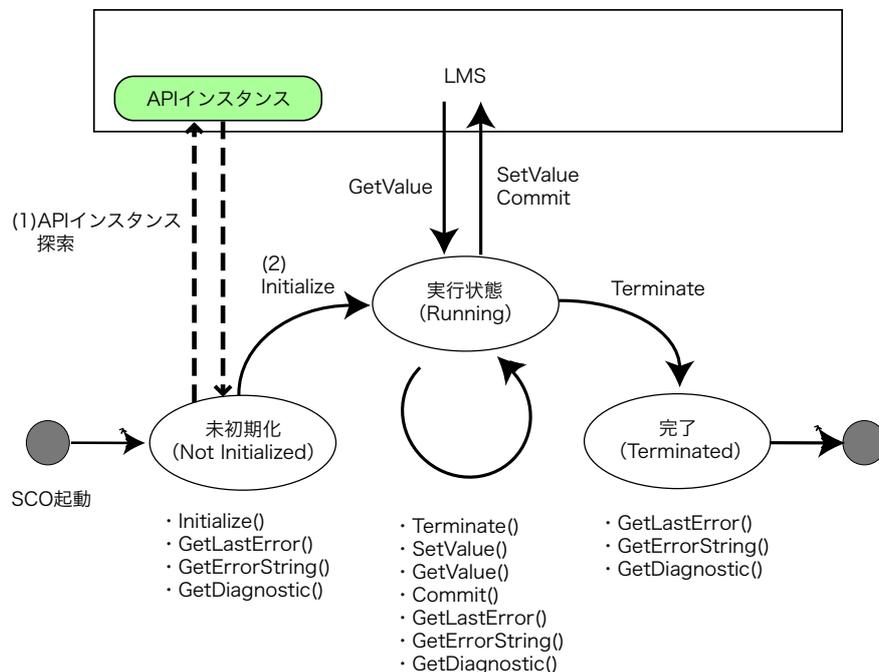


図 3.1 ランタイム環境における SCO と LMS の通信メカニズム

3.1 SCORM のデータモデル

SCORM のデータモデルとは、SCO と LMS の間でやり取りできるデータのことを指す。このデータには、演習問題に対する学習者の回答や得点、学習時間などがある。LMS では

3.1 SCORM のデータモデル

これらのデータを学習管理に役立てている。しかし、データモデルには、それぞれに LMS に対する読み込み・書き込みの制限や、データ型に違いがあるなど複雑である。

3.1.1 データ型の種類

データモデル要素ごとに、扱うデータ型が異なる。以下に、データモデルが扱うデータ型を示す。

- キャラクタ文字列

キャラクタ文字列とは、ISO 10646 (Unicode 標準に相当) において定義されている文字列のことである。

- ローカル文字列

ローカル文字列とは、言語の指定を要求する文字列である。特に指定がない限りはデフォルトで英語が指定される。

- 言語タイプ

このデータ型は、言語を指定するために使用され、文字列で表現される。

- 長い識別子タイプ

このデータ型は、URI (Uniform Resource Identifier) に従った文字列である。SCORM では、この URI が URN (Uniform Resource Name) の形式で表現され、グローバルな範囲でユニークな文字列であることが推奨されている。これは、識別子に関するデータモデル要素において用いられる。

- 短い識別子タイプ

このデータ型は、SCO 内部においてユニークな文字列で記述される。長い識別子タイプとの違いは、このデータ型はグローバルな範囲で利用されることを想定していないということである。

- 整数 (integer)

整数は、コレクションの数などを示す場合に用いられる。

3.1 SCORM のデータモデル

- 状態 (state)

このデータ型は、データモデル要素ごとに決められた文字列で SCO 学習の状態を表現している。例えば、学習の合否状態を表す `cmi.success_status` には “passed”, “failed”, “unknown” という状態がある。

- 実数 (real)

実数は、有効数字 7 桁で表現される。例えば、正規化された値を表現する際に用いられる。

- time

time とは、秒単位で表現されるデータ型のことである。

- timeinterval

timeinterval は、経過時間を表すためのデータ型である。例えば、“P1Y3M2DT3H” (= 1 年 3 ヶ月と 2 日と 3 時間) と表現される。

3.1.2 SPM

データモデル要素の中には、SPM (Smallest Permitted Maximums) が定義されているものがいくつかある。SPM とは、最低限保証される最大値を意味しており、これは文字列の文字数あるいはコレクションを有するデータモデルのコレクション数として定義されている。

3.1.3 マニフェストファイルの記述内容で設定するデータモデル要素

XML 形式で記述されるマニフェストファイルを参照して、その値を SCO の初期化データとして用いるデータモデルがある。そのデータモデル要素をを表 3.1 に示す。

3.1 SCORM のデータモデル

3.1.4 コレクションとして定義されるデータモデル要素

データモデルの中には、コレクションとして定義されるものがある。これらは、配列のよ
うに 0 から数えはじめるインデックスを持っている。以下にそのデータモデルを示す。

- cmi.comments_from_learner
学習者からのコメントに関するデータモデル要素である。
- cmi.comments_from_lms
LMS からのコメントに関するデータモデル要素である。
- cmi.objectives
SCO を通じた学習活動に対する学習目標に関するデータモデル要素である。
- cmi.interactions
SCO 学習における学習者からの応答に関するデータモデルである。ここでいう応答と

表 3.1 マニフェストファイルの記述内容で設定するデータモデル要素

データモデル要素名	説明
マニフェスト要素名	
cmi.completion_threshold	SCO を完了したとみなす ために必要な進捗度
< adlcp:completionThreshold >	
cmi.launch_data	SCO 起動時の 初期化パラメータ
< adlcp:dataFromLMS >	
cmi.max_time_allowed	SCO の最大許容実行時間
< imsss:limitConditions attemptAbsoluteDurationLimit >	
cmi.scaled_passing_score	SCO の合格点となる 正規化された閾値
< imsss:minNormalizedMeasure >	
cmi.time_limit_action	制限時間経過後の SCO の動作
< adlcp:timeLimitAction >	

3.2 API インスタンス

は、例えば、演習の回答などがあてはまる。

図 3.2 において、ドット “.” で区切られた n あるいは m という文字がインデックスとなっており、ここに 0 から数え始める正の整数が入る。例えば、cmi.interactions.2.id は 3 番目のインタラクション識別子を意味している。インデックスの最大数は、データモデル要素ごとの SPM で決まる。

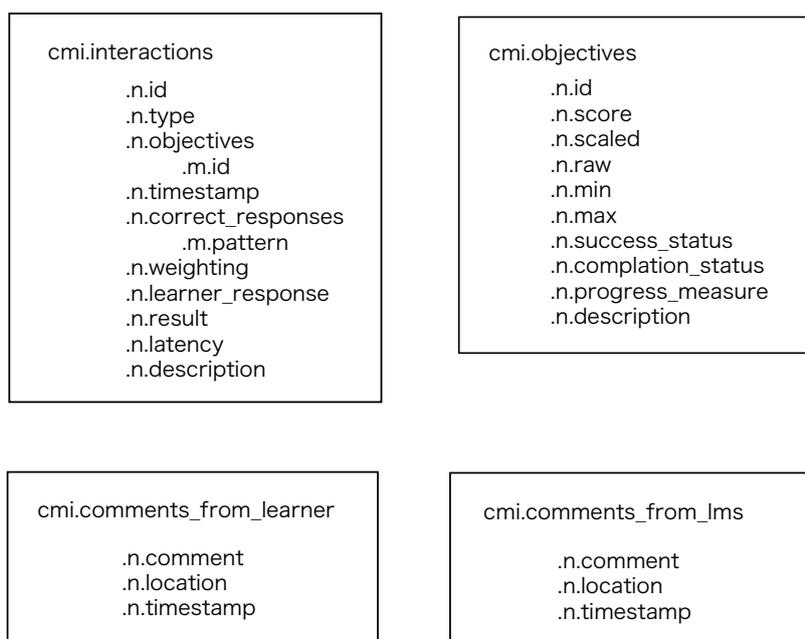


図 3.2 レコードタイプのデータモデル

3.2 API インスタンス

API インスタンスとは、SCO と LMS の間で通信を行う際に用いられる関数の集まりを表す。これは、LMS によって提供され、SCO 側に JavaScript で記述して用いる。一般に、SCORM 対応 LMS とは、SCO に対して提供する API インスタンスを実装しているものを指す。API インスタンスは、大きく分けると、データ転送関数、通信制御関数、エラー関数の 3 種類に分けられる。

3.2 API インスタンス

3.2.1 API_1484_11

“API_1484_11”は、SCORM2004 における API インスタンスの名称である。LMS 側では、この名称で SCO がアクセスできるようにしなければならない。また SCO 側では、まずこのオブジェクトを探し出し API インスタンスを利用できる状態にしなければならない。

3.2.2 API 探索アルゴリズム

SCO と LMS がデータをやり取りするために、まず SCO は、LMS 側に用意されている API_1484_11 を探し出す必要がある。この探索のアルゴリズムの実装は SCO 側、すなわちコンテンツ作成者に任されているが、図 3.3 で示すように、SCO を開いたウィンドウから連鎖している親ウィンドウに対して再起的に探索を行わなければならない。

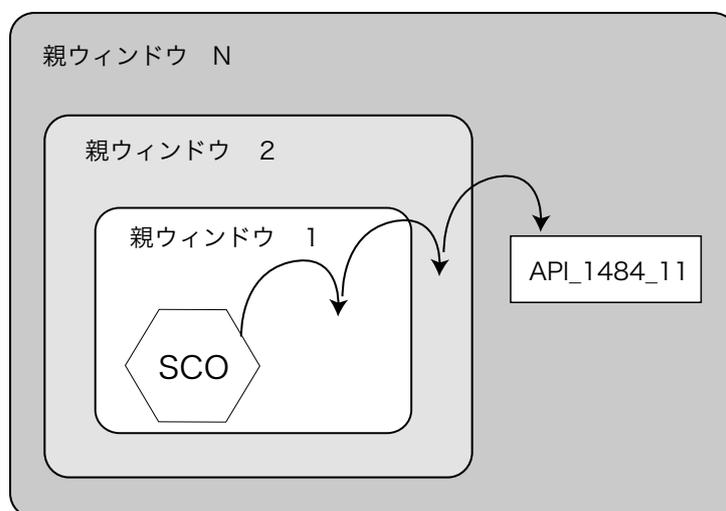


図 3.3 API インスタンスの探索例

3.2.3 データ転送関数

データ転送関数は、SCO 学習中のプロセスにおいて、データモデルを引数に用いて LMS と通信を行う関数である。SCO において、この関数の利用は任意である。以下にデータ転送関数を述べる。

3.2 API インスタンス

- GetValue(“データモデル要素”)

GetValue は LMS からデータを取得する際に用いられる関数である。引数には読み込み可能なデータモデルを用いることができる。JavaScript の記述において、任意の変数に代入する形で利用される。

- SetValue(“データモデル要素”, “送信する値”)

SetValue は、LMS へのデータ送信の際に用いられる関数である。JavaScript の記述において、引数に利用するデータモデル要素と、LMS に書き込む値を指定する形で利用される。LMS が提供する API インスタンスの実装により、一旦データをキャッシュして送信するか、直接 LMS に送信するかが異なっている。

- Commit(“”)

Commit は、SetValue で一度キャッシュに蓄えられたデータを LMS に送信する際に用いられる。

3.2.4 通信制御関数

通信制御関数は、通信の初期化あるいは終了を制御する関数である。従って、SCO 内においてこれらの関数は、必ず呼び出さなければならない。以下にその関数を述べる。

- Initialize(“”)

Initialize は、SCO と LMS の間における通信の初期化を行う際に用いられる。LMS より API インスタンスを発見次第、この関数により通信が初期化される。SCO 側では、このように動作するように JavaScript で記述する必要がある。

- Terminate(“”)

Terminate は、SCO と LMS の間における通信を終了する際に用いられる関数である。この関数が呼び出された後は、エラー関数以外呼び出すことは不可能である。

3.2 API インスタンス

3.2.5 エラー関数

エラー関数は、ランタイム環境における一連の処理、すなわち SCO 起動から終了までのプロセスにおいてエラーが発生した際に、そのエラーに関する情報を提示する等の処理を行う関数である。以下にその関数を述べる。

- GetLastError()

ランタイム環境の一連の処理内で発生したエラー情報を LMS に対して要求する関数である。戻り値として、発生した最新エラーのエラーコードが文字列で返される。

- GetErrorString(“エラーコード”)

エラーコードを引数に用いることで、そのエラーコードに対応した文字列を返す。

- GetDiagnostic(“エラーコード等”)

LMS が個別利用するための出力関数である。また、診断情報の付加も可能である。

第 4 章

SCORM コンテンツ作成支援システムの設計

この章では、SCORM コンテンツ作成支援システムを構築するためのシステム設計方法について述べる。

4.1 システム構築の目的

SCORM コンテンツ作成支援システム構築の目的は、SCORM に関する知識や必要なファイルの記述方法を知らないコンテンツ作成者でも、SCORM に準拠したコンテンツを容易に作成できるように支援することである。

4.2 作成対象となる SCORM コンテンツ

本研究で作成の対象とする SCORM コンテンツは、SCORM 規格の中でも最も多様なデータをやりとりできる SCORM2004 に準拠させる。さらにデータモデル及び API インスタンスの利用という観点から、テスト・演習用のコンテンツを作成対象とする。テスト・演習用のコンテンツは、閲覧用のコンテンツとは異なり、インタラクション (cmi.interactions) に関するデータモデルを利用できる。インタラクションに関するデータモデルとは、演習やテストで学習者が答えた回答や正誤判定の結果などを LMS に学習履歴情報として保存するために用いられるデータモデルのことである。このインタラクションのデータモデルの中に、“cmi.interactions.n.type” というデータモデル要素がある。これには決まった値空間が

4.3 支援方法

存在し、回答形式及び LMS への学習結果の反映に影響を与える。表 4.1 に示された値空間で回答情報を表現できるのであれば、問題の表現方法や回答形式はコンテンツ作成者の自由である。

作成可能な問題の種類は表 4.1 より、正誤問題、単一選択問題、複数選択問題、記述問題、並べ替え問題、組み合わせ問題、穴埋め問題の 7 種類とする。

表 4.1 インタラクションのデータのタイプ

cmi.interaction.n.type の値空間 (状態)	説明	本システムで作成できる問題
true-false	正誤 (○×)	○
choice	選択 (単一・複数)	○
fill-in	穴埋め	○
long-fill-in	記述 (完全一致)	○
likert	アンケート	
matching	組み合わせ	○
performance	パフォーマンス測定	
sequencing	並べ替え	○
numeric	数値	
other	その他	

4.3 支援方法

コンテンツ作成者が SCORM コンテンツを作成するには、教材作成の他にコンテンツアグリゲーションモデルとランタイム環境を実現させる必要がある。そして、それらを実現するために XML、JavaScript を記述しなければならない。さらにこの作業を行うには、SCORM に関する複雑な知識が要求される。従って、教材の作成に専念できないコンテンツ作成者もいる。

本研究で構築する SCORM コンテンツ作成支援システムでは、SCO そのものである

4.4 システムの動作

HTML ファイル、コンテンツアグリゲーションモデルを実現する XML (マニフェスト) ファイル、ランタイム環境を実現する JavaScript ファイルを自動で作成することで SCORM コンテンツ作成の支援を行う。コンテンツ作成者に求める作業は、問題の作成など教材作りに必要とされるに限定する。

4.4 システムの動作

本システム内部では、ループ処理によりテンプレートファイルに記述された文字列を一行ずつ読み込んでいる。各テンプレートファイルには、それぞれにコメントアウトされた特殊な文字列が記述されており、この文字列が読み込まれた際に、コンテンツ作成者からの入力データを書き込む処理を行っている。このようにして、テンプレートファイルの文字列に入力データの文字列を追加してできた文字列を、新しいファイルとして出力することでそれぞれのファイルが作成される。そして、作成された 3 つのファイルは、コンテンツ作成者が入力したフォルダ名で作成されたフォルダに出力される。

4.5 テンプレートファイル

本システムが用いるテンプレートファイルは、最低限必要なタグや関数などが予め記述されているテキスト形式のファイルである。これに対し、コンテンツ作成者からの入力情報を適した場所に記述している。

4.5.1 HTML のテンプレートファイル

HTML のテンプレートファイルには、“< HEAD >”などの HTML 必須タグ、回答終了のボタン、JavaScript ファイルを利用するためのタグが記述されている。さらに、コンテンツ作成者からの入力情報を HTML ファイルに記述するためのコメントアウトされた文字列が記述されている。それらの詳細について以下に述べる。

4.5 テンプレートファイル

- <!--que-->

問題番号、問題文、回答入力フォームを記述するために用いられる。

- <!--lans-->

学習者の回答を配列に格納する処理を記述するために用いられる。

- <!--time-->

制限時間を設けて、かつ制限時間経過後に学習を強制終了させる場合、この文字列の次の行にその処理を記述するために用いられる。

- <!--seq-->

並べ替え問題を作成する際に、必要となる関数を記述するために用いられる。その関数は、ボタンに書かれた回答要素の文字列を空白のテキストに記述する処理を行う。

4.5.2 XML のテンプレートファイル

XML で記述されるマニフェストファイルには、以下のコメントアウト文字列を用いて入力情報の記述を行う。

- <!-- id-->

SCO の識別子を記述するために用いられる。入力された文字列と固定された文字列“_B1”を連結させて識別子としている。

- <!-- title-->

入力された SCO のタイトルを記述するために用いられる。

- <!-- comTh-->

SCO を完了とみなす閾値を記述するために用いられる。入力された閾値 (%) を 0 ~ 1 の範囲の実数で正規化して記述する。本システムでは、この値と合格閾値 (<imsss:minNormalizedMeasure >) に同じ値を設定している。

- <!-- tiLim-->

設定された制限時間を経過した後の SCO の振る舞いを記述するために用いられる。

4.5 テンプレートファイル

SCORM の規格書によると、強制終了あるいは学習継続の際にメッセージを残すことが可能である。しかし、本システムはメッセージは残さない仕様である。

- <!-- limCo -->

SCO の学習に費やせる最大許容時間を記述するために用いられる。本システムでは、秒単位で入力された制限時間を timeinterval として表現されるデータ型に変換して記述している。

- <!-- minNM -->

合格閾値を記述するために用いられる。これは、入力された合格閾値 (%) を 100 で割って算出される 0~1 の範囲で表された値が記述される。また、この値は SCO の完了を判定する閾値にも利用される。

4.5.3 JavaScript のテンプレートファイル

JavaScript のテンプレートファイルには、各データモデル要素のパラメータとなる変数や、API インスタンス探索、通信の初期化あるいは終了、データのやりとり、合否判定を行う関数などが記述されている。

- //fs=*

各設問に対する配点の合計点を変数に記述するために用いられる。この変数は、合否判定と学習者得点の正規化に利用される。

- //ic=*

設問数をカウントする変数に、作成された問題数を代入するために用いられる。

- //sps=*

値の取得に失敗したことを想定して、暫定的に合格閾値を変数に代入しておくために用いられる。

- //icrc=*

各設問ごとの正解数を配列に格納するために用いられる。

4.6 SCORM コンテンツを構成するファイル

- `//=ity=*`

インタラクションのタイプ、すなわち出題形式のタイプを記述するために用いられる。
このタイプとは表 5.1 に示す値で表現される状態のことである。

- `//=corp=*`

各設問ごとに入力された正解パターンをすべて記述するために用いられる。

- `//=ans=*`

テンプレートファイルに予め記述されている“ansQ”という関数を問題の数だけ呼び出すための記述を行う。この関数は正誤判定に用いられ、引数として、0 から数えた問題番号、問題文、配点、出題形式（インタラクションのタイプ）の 4 つが渡される。

4.6 SCORM コンテンツを構成するファイル

本システムにより作成される SCORM コンテンツは、HTML ファイル、XML ファイル、JavaScript ファイルから構成されている。JavaScript ファイルと HTML ファイルは一つにまとめることも可能であるが、学習者がソースの閲覧を行うことによるテストの不正行為をさせないためにこれらは別々に記述した。以下に、それぞれのファイルについて説明する。

- HTML ファイル

HTML ファイルは、学習資源となる SCO そのものである。このファイルには問題文、回答形式に合わせたフォーム、学習者からの回答情報を収集する処理を行う JavaScript が記述されている。

- XML ファイル

XML ファイルは、メタデータなど、コンテンツアグリゲーションモデルの定義で必要とされている記述要素をはじめ、制限時間や SCO 合格点などのデータモデルが参照して初期化するための値などが記述されているマニフェストファイルのことである。ファイル名は、SCORM 規格の定義に従い“imsmanifest.xml”としている。

- JavaScript ファイル

4.6 SCORM コンテンツを構成するファイル

JavaScript ファイルには、API 探索、データモデルや API インスタンスなど、ランタイム環境で必要とされている処理の内容、さらに、SCO の合否判定や各問題の正誤判定などの処理も記述されている。ファイル名に関しては SCORM 規格で定義されていないが “lms.js” としている。

第 5 章

SCORM コンテンツ作成支援システムの構築

第 2 章 (2.5 節) で述べた問題点を改善するために、本研究では、JavaScript などの記述方法や、SCORM に関する知識を持たないコンテンツ作成者でも容易に SCORM に準拠したコンテンツを作成できる SCORM コンテンツ作成支援システムを構築する。

5.1 システムの概要

本システムの構築には Java 言語を用いた。コンテンツ作成者は、問題の考案など従来の教材作りに必要とされる知識のみでテスト・演習用の SCORM コンテンツを作成できるようにした。

コンテンツ作成者は、以下に示す順にデータの入力を行う。また、図 5.1 はそれを図で説明したものである。

1. ファイル格納フォルダ名と SCO 名の入力
2. 問題の作成
3. 各問題の配点、制限時間、制限時間後の SCO の動作の設定

5.1 システムの概要

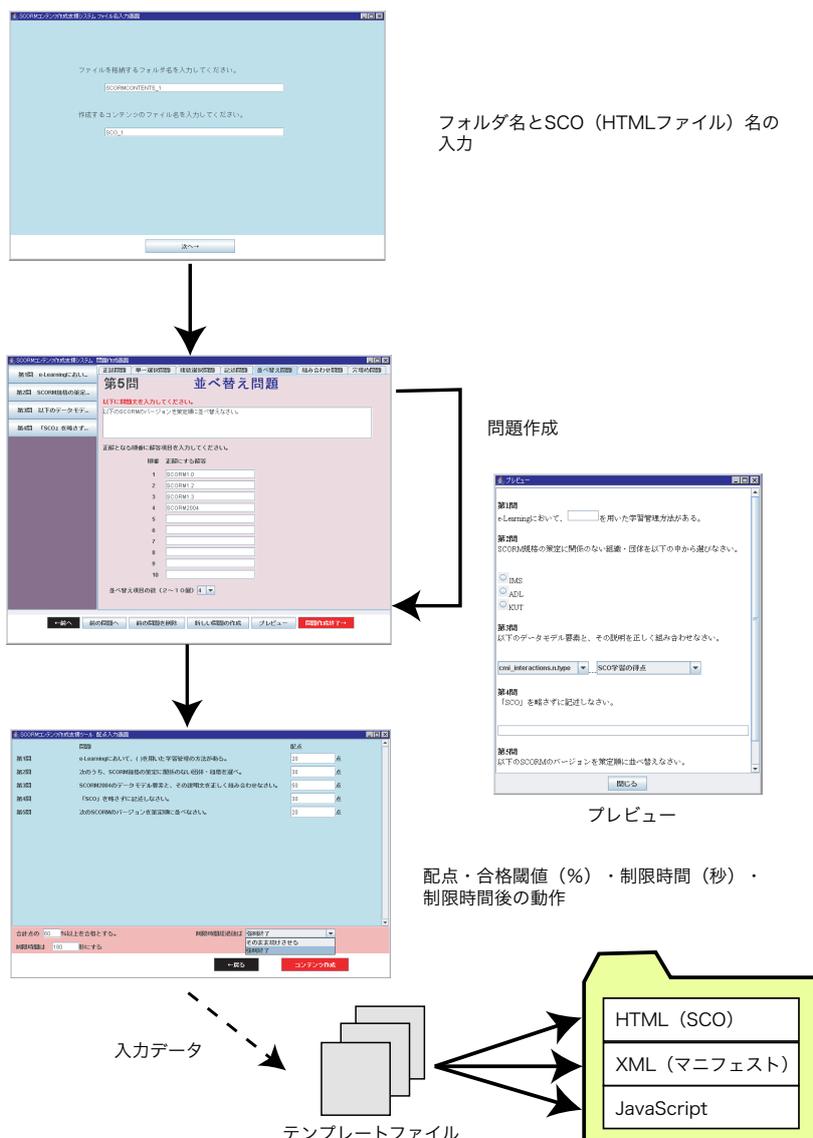


図 5.1 コンテンツ作成までの画面遷移

5.1.1 フォルダ名・ファイル名の入力

コンテンツ作成者は、まず最初に表示される画面で SCO 名とフォルダ名の入力を行う必要がある。ここで入力された SCO 名は、マニフェストファイルにおけるタイトル及び識別子となる文字列の一部としても用いられる。

5.1 システムの概要

5.1.2 作成可能な問題

ここでは、本システムを用いて作成可能な7種類の問題に関して説明する。なお、SPMの制限により、作成可能な問題数を最大250問、正解パターン数を最大10個までとした。

正誤判定問題

正誤判定問題は、文章の内容に対する正誤を問う問題である(図5.2)。コンテンツ作成者の入力事項は次の通りである。

- 正誤判定の対象となる問題文
- 問題文(2種類のうちから選択)
 - － 次の文章が正しければ○を、誤っているなら×を選びなさい。
 - － 次の分に誤りを見つけたら○を、誤りがなければ×を選びなさい。
- ○か×のうち正解である方の選択。

学習者は、この問題に対してラジオボタンへチェックを入れることで回答する。

単一選択問題

単一選択問題は、与えられた分岐要素に対して一つだけ選択させる問題である(図5.2)。この問題に関するコンテンツ作成者の入力事項は次の通りである。

- 問題文
- 選択肢
- 正解の選択肢(ラジオボタンにチェックを入れる)
- 選択肢の数(最大10択まで)

この問題は、コンテンツ化の際に記入された順番をシャッフルして表示順を変えることが可能である。学習者は、この問題に対してラジオボタンへチェックを入れることで回答する。

5.1 システムの概要

複数選択問題

複数選択問題は、与えられた分岐要素に対して、複数の選択をさせる問題である（図 5.2）。この問題に関するコンテンツ作成者の入力事項は次の通りである。

- 問題文
- 選択肢
- 正解の選択肢（チェックボックスにチェックを入れる）
- 選択肢の数（最大 10 択まで）

この問題は、コンテンツ化の際に記入された順番をシャッフルして表示順を変えることが可能である。学習者は、この問題に対してチェックボックスへチェックを入れることで回答する。

記述問題

記述問題は、一つの単語以上から成る回答をさせる問題である（図 5.2）。コンテンツ作成者によって決められた正解パターンを用いて回答情報と照合を行い正誤を判定する。従って回答とパターンが完全に一致しないと正解とはみなさない。この問題に関するコンテンツ作成者の入力事項は次の通りである。

- 問題文
- 正解のパターン
- 正解パターン数（最大 10 個まで）

学習者は、この問題に対して与えられたテキスト入力フォームへの直接入力での回答する。

5.1 システムの概要

並べ替え問題

並べ替え問題は、与えられた要素に対して、それを正しい順番に並べる問題である（図 5.2）。この問題に関するコンテンツ作成者の入力事項は次の通りである。

- 問題文
- 並べ替え要素
- 並べ替え要素数

並べ替え要素の入力は、1 番目の入力欄から正解順に入力する必要がある。この問題の並べ替え要素は、コンテンツ化の際に自動でシャッフルされて表示される。学習の際、学習者が表示されたボタンをクリックすることで、そのボタンに記述された文字列が空のテキスト入力フォームに表示される。このテキストに表示された文字列が学習者の回答情報となり正誤判定に用いられる。

組み合わせ問題

組み合わせ問題は、用意された二つのプルダウンメニューの中から選択して組み合わせで回答させる問題である（図 5.3）。この問題に関するコンテンツ作成者の入力事項は次の通りである。

- 問題文
- 問題群とそれに対応した回答群
- 回答のペア数（最大 10 組まで）

なお、問題群と回答群の正解となる組み合わせは 1 対 1 に対応している。

5.1 システムの概要

穴埋め問題

穴埋め問題は、一つの単語で空欄を補う問題である（図 5.3）。この問題に関するコンテンツ作成者の入力事項は次の通りである。

- 回答欄の直前にくる問題文
- 正解のパターン
- 回答欄の直後にくる問題文
- 正解パターン数（最大 10 個まで）

問題文の入力は、回答欄の直前あるいは直後のみでもよい。学習者は、この問題に対して与えられたテキスト入力フォームへの直接入力で回答する。

5.1.3 その他の入力事項

コンテンツ作成者は、問題作成終了後に表示される画面で各設問の配点、合格閾値、制限時間及びその経過後の SCO の動作を決めるためのデータ入力を行う必要がある。ここでは、その各入力事項について述べる。

- 配点

問題作成終了後に表示される画面には作成したそれぞれの問題文が表示され、その隣に配点入力用のフォームが表示される。ここにそれぞれの配点を入力する。もし、フォームに配点が未入力の場合は、その設問の配点は自動的に 0 点となる。

- 合格閾値

合格閾値とは、各設問の配点を足し合わせた合計点に対して何%以上の得点で合格とみなすかの閾値のことである。コンテンツ作成者は、0~100（%）の範囲で値を入力する。この値は、学習完了の閾値としても用いられる。

- 制限時間

制限時間は秒単位で入力を行う。この値は、制限時間超過を知らせるアラートを表示さ

5.1 システムの概要

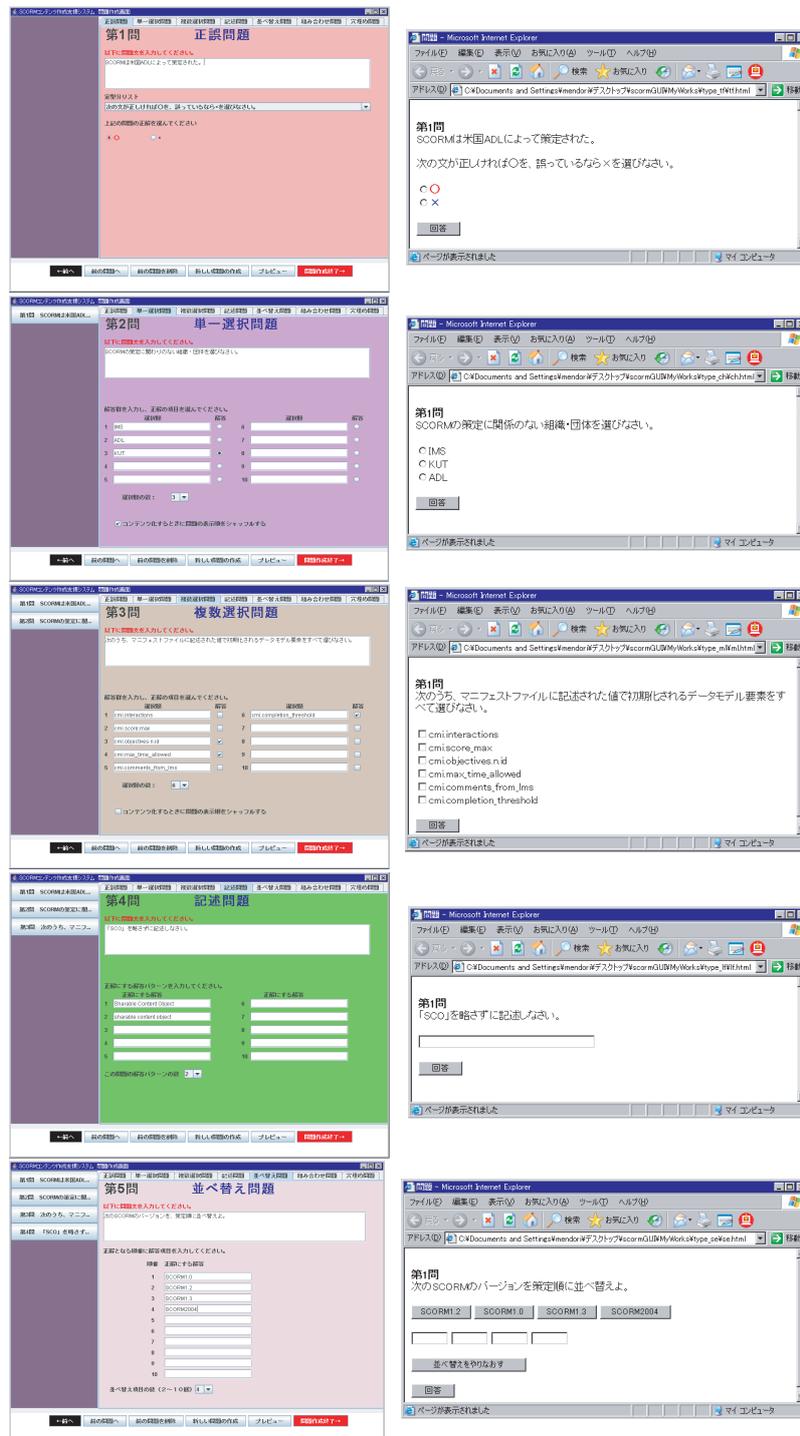


図 5.2 問題作成画面 (左) と学習画面 (右)

5.2 動作確認

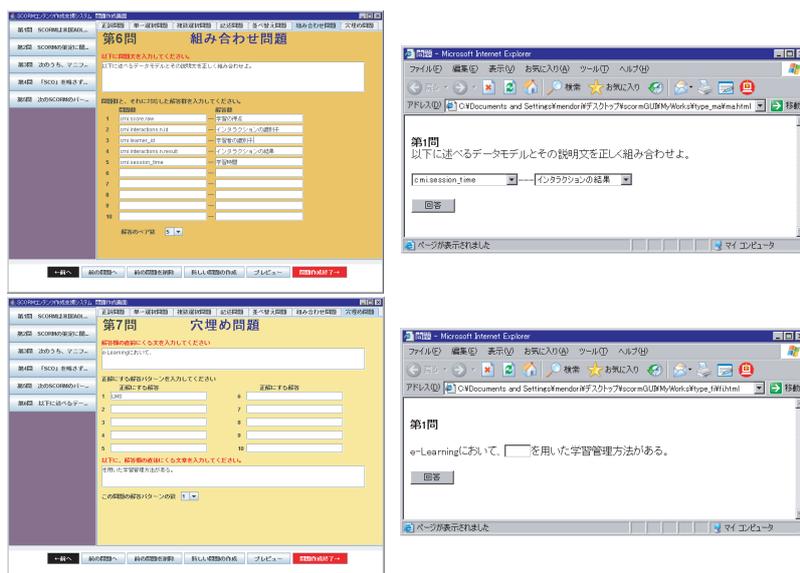


図 5.3 問題作成画面と学習画面（続き）

せるために JavaScript に記述される。またこの値は、timeinterval として表現される文字列に変換して、マニフェストファイルに対して記述される。

- 制限時間後の SCO の動作

制限時間超過後に学習を強制終了するか、あるいは継続するかを決める。これはデータモデル要素である `cmi.time_limit_action` の値（状態）を決めるために選択する。「継続」を選んだ場合、マニフェストファイルにこのデータモデル要素の値である “continue,no message” が記入され、「強制終了」を選んだ場合は同様に “exit,no message” が記入される。

5.2 動作確認

本システムを用いて作成されたコンテンツが、実際に SCORM コンテンツとして機能しているか確認作業を行った。その方法として、まずは SCORM2004 対応の LMS にコンテンツの登録を行った。今回は、動作確認用の LMS として、NTT-AT が無料で公開している OpenSourceLMS を利用した。次に、登録したコンテンツの学習を試行した（図 5.4）。そして、学習ログを参照して学習結果が LMS に反映されているかどうか確認した（図 5.5）。そ

5.2 動作確認

の結果、得点や回答情報、合否の結果などの情報が LMS に反映していることが確認できた。

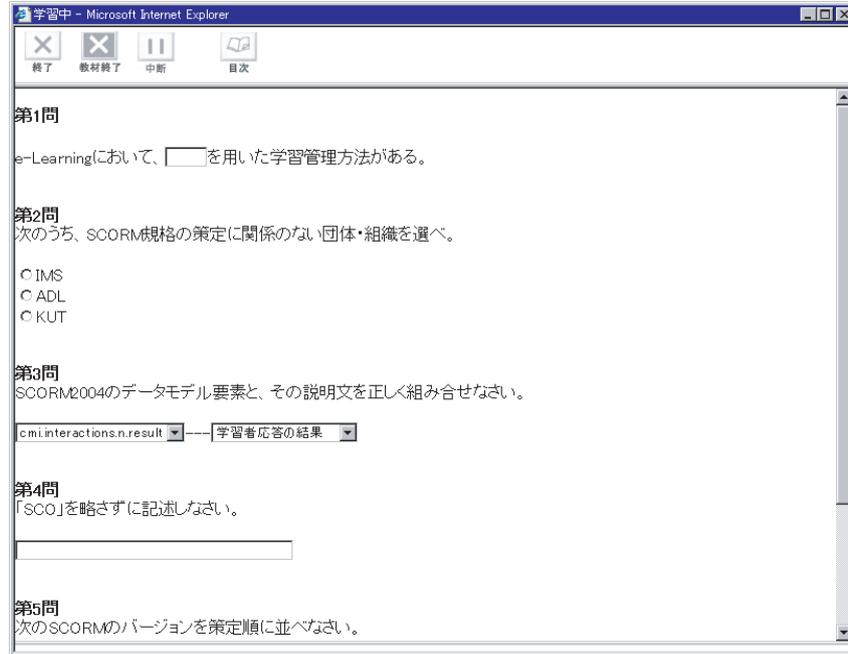


図 5.4 LMS から起動した SCO の学習画面

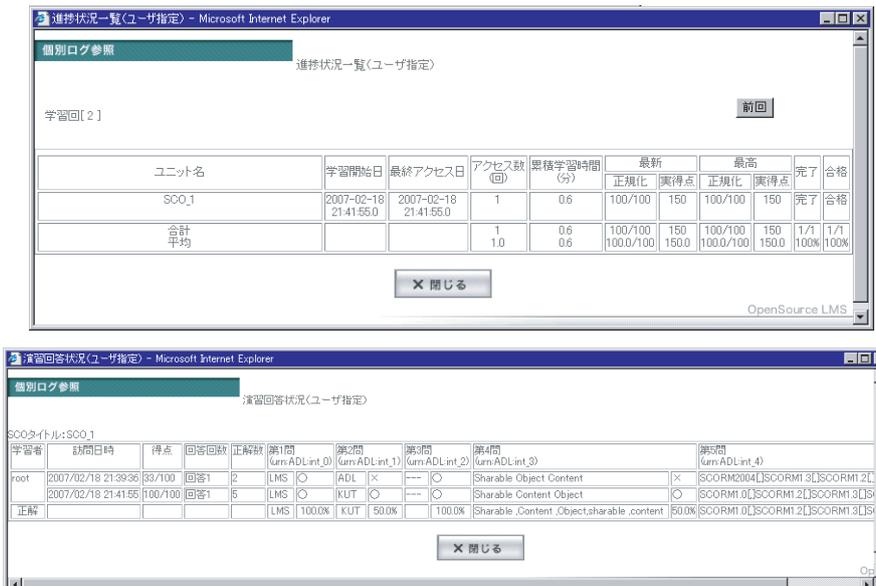


図 5.5 学習進捗ログ表示画面（上）と演習回答ログ表示画面（下）

第 6 章

おわりに

本研究において、SCORM2004 のデータモデル及び API インスタンスの利用に着目して、演習・テスト用の SCORM コンテンツを作成できるシステムを構築した。本システムは、本来なら SCORM コンテンツ作成者が行うべき作業である HTML、XML、JavaScript の記述を自動的に行うことで、コンテンツ作成者の SCORM コンテンツ作成を容易にした。また本システムは、コンテンツ作成者が SCORM に関する複雑な知識を持っていなくても、問題文や正解パターンの入力を行うだけで SCORM コンテンツを作成できるため、教材作りに専念できるようになった。

しかし、本システムは、現段階では一度にひとつの SCO だけを作成する仕様になっているため、複数の SCO にまたがって適用されるシーケンシングやナビゲーションは適用されない。これらを適用すれば、学習者一人一人の学習進捗をより高度に管理できるようになること、及び学習者の学習効果を高めることが期待できる。そこで、今後の課題として、本システムを複数の SCO から構成される SCORM コンテンツの作成を可能にすること、また、それに伴いシーケンシングやナビゲーションの制御ができるように改良することが挙げられる。

謝辞

本研究全般におきまして、親身になって多大なるご指導を賜りました高知工科大学情報システム工学科の妻鳥貴彦講師には心からの感謝の意を表します。

また、本研究の副査を引き受けて頂きました同学科の篠森敬三教授には心から御礼申し上げます。

同じく、本研究の副査を引き受けて頂きました同学科の島村和典教授には心から御礼申し上げます。

そして、SCORM に関する様々な知識を与えてくださった妻鳥研究室 4 回生の寒川剛志氏、SCORM コンテンツ作成支援システムの実装に協力してくださった同研究室 3 回生の畠山博和氏、ご丁寧に本稿の添削をしてくださった高知工科大学大学院妻鳥研究室 1 回生の高木翔平氏、LMS サーバ構築の際に多大なご助力を頂いた本学大学院同研究室 1 回生の木下聡氏に深く感謝致します。

参考文献

- [1] Advanced Distributed Learning, “SCORM 2004 3rd EDITION,” 2006/10/20
- [2] 特定非営利活動法人日本イーラーニングコンソシアム, “SCORM 2004 解説書 第 1.0.4 版”, 2006 年 3 月
- [3] 特定非営利活動法人日本イーラーニングコンソシアム, “SCORM 2004 コンテンツ作成ガイド 第 1.0.4 版”, 2006 年 3 月

付録 A

データモデル概要

表 A.1 SCORM2004 のデータモデル要素の概要

データモデル要素	データ内容
cmi.comments_from_learner	学習者からのコメント
cmi.comments_from_lms	LMS からのコメント
cmi.completion_status	SCO 学習の完了状態
cmi.completion_threshold	SCO 学習を完了とみなすための閾値
cmi.credit	SCO 学習における学習者のパフォーマンスに対する評価
cmi.entry	SCO にアクセスしたかどうかを示す状態
cmi.exit	SCO から退出した方法
cmi.interaction	問題の回答などの学習者応答
cmi.launch_data	SCO の起動に用いるデータ
cmi.learner_id	学習者の識別子
cmi.learner_name	学習者の氏名
cmi.learner_preference	学習者ごとの個別設定情報
cmi.location	SCO の格納場所
cmi.max_time_allowed	SCO 学習の最大許容時間
cmi.mode	SCO の動作モード
cmi.objectives	学習目標

表 A.2 SCORM2004 のデータモデル要素の概要 (続き)

データモデル要素	データ内容
cmi.progress_measure	SCO 学習進捗の測定値
cmi.scaled_passing_score	SCO 合格点の正規化表現
cmi.score	SCO に対する学習者の得点
cmi.session_time	SCO 学習に費やした時間
cmi.success_status	SCO の合否状態
cmi.suspend_data	SCO 中断時に保存しておく情報
cmi.time_limit_action	タイムリミット後の SCO の動作
cmi.total_time	SCO の総合学習時間