

| | |
|--------------|---|
| Title | ナビゲーション機能を有する学習管理システムに関する研究 |
| Author(s) | 阿部, 博 |
| Citation | |
| Issue Date | 2003-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/1661 |
| Rights | |
| Description | Supervisor:落水 浩一郎, 情報科学研究科, 修士 |

修士論文

ナビゲーション機能を有する
学習管理システムに関する研究

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

阿部 博

2003年3月

修 士 論 文

ナビゲーション機能を有する
学習管理システムに関する研究

指導教官 落水浩一郎 教授

審査委員主査 落水浩一郎 教授

審査委員 丹康雄 助教授

審査委員 権藤克彦 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

110004 阿部 博

提出年月: 2003 年 2 月 14 日

概要

本論文では、学習者が遠隔学習システムを利用する際に、学習者の目的にあった適切な学習項目の選択を支援することを目的として、ナビゲーション機能を有する学習管理システムを提案する。また、ナビゲーション機能を実現する上で、学習管理システムにおける標準規格となりつつある SCORM で定義される情報を効果的に利用し、さらに拡張性、汎用性を考慮したシステムの枠組みを示す。また、本システムを実現するために改良した本学遠隔学習システムの概要について報告する。

目次

| | | |
|-------|------------------|----|
| 第1章 | はじめに | 1 |
| 1.1 | 背景 | 1 |
| 1.2 | 目的 | 1 |
| 1.3 | 本論文の構成 | 3 |
| 第2章 | ナビゲーション機能の必要性 | 4 |
| 2.1 | 本学遠隔学習システムについて | 4 |
| 第3章 | 学習管理システム (LMS) | 7 |
| 3.1 | 学習管理システムとは? | 7 |
| 3.2 | 学習管理システムにおける標準規格 | 10 |
| 第4章 | SCORM | 12 |
| 4.1 | 概要 | 12 |
| 4.2 | コンテンツ集約モデル | 14 |
| 4.3 | コンテンツモデル | 14 |
| 4.3.1 | Asset | 15 |
| 4.3.2 | SCO | 15 |
| 4.3.3 | コンテンツアグリゲーション | 16 |
| 4.4 | メタデータ | 16 |
| 4.5 | コンテンツパッケージング | 17 |
| 4.6 | データモデルとAPI | 18 |
| 4.6.1 | データモデル | 18 |
| 4.6.2 | API | 18 |
| 4.7 | ナビゲーション機能に必要な情報 | 19 |
| 4.7.1 | 教材情報 | 19 |
| 4.7.2 | 学習者情報 | 19 |
| 第5章 | ナビゲーション機能の実現 | 21 |
| 5.1 | 学習状態モジュール | 21 |
| 5.2 | ナビゲーション戦略モジュール | 23 |

| | | |
|------------|---------------------|-----------|
| 5.3 | ナビゲーション機能 | 26 |
| 第6章 | システムの概要 | 29 |
| 6.1 | システムの枠組み | 29 |
| 6.1.1 | サーバ環境 | 29 |
| 6.1.2 | クライアント環境 | 29 |
| 6.2 | XML ファイル作成モジュール | 32 |
| 6.3 | XML データベース | 34 |
| 第7章 | ナビゲーションの実行 | 36 |
| 7.1 | 学習を進める順序 | 36 |
| 第8章 | 本システムの改善に向けて | 43 |
| 8.1 | ナビゲーションの単位 | 43 |
| 8.2 | ユーザインタフェース | 43 |
| 8.3 | 学習者履歴の活用 | 43 |
| 8.4 | 教材の分散管理 | 44 |
| 8.5 | ナビゲーション機能のライブラリ化 | 44 |
| 第9章 | おわりに | 46 |
| 9.1 | まとめ | 46 |
| 9.2 | 今後の課題 | 46 |
| 9.2.1 | 本システムの改善 | 46 |
| 9.2.2 | 実証実験 | 46 |
| | 謝辞 | 47 |
| | 参考文献 | 48 |
| | 付録 | 51 |

目 次

| | | |
|-----|---------------------------|----|
| 2.1 | 本学の遠隔学習システム | 4 |
| 2.2 | index 利用時における関連項目検索の難しさ | 5 |
| 3.1 | 従来の学習管理システムのイメージ | 9 |
| 3.2 | 本研究で開発する学習管理システムのイメージ | 10 |
| 4.1 | コンテンツアグリゲーション | 15 |
| 4.2 | データモデル | 18 |
| 5.1 | 学習状態モジュール | 22 |
| 5.2 | 学習状態モジュール群 | 22 |
| 5.3 | ナビゲーション戦略モジュールと学習状態モジュール群 | 23 |
| 5.4 | ナビゲーション戦略モジュール1 | 24 |
| 5.5 | ナビゲーション戦略モジュール2 | 25 |
| 5.6 | ナビゲーション戦略モジュール3 | 25 |
| 5.7 | ナビゲーション戦略モジュール4 | 26 |
| 5.8 | ナビゲーション機能 | 27 |
| 6.1 | システムイメージ | 30 |
| 6.2 | クライアントイメージ | 31 |
| 6.3 | 生成されたXML ファイル | 33 |
| 6.4 | Xindice における文章管理 | 35 |
| 7.1 | ログイン画面 | 37 |
| 7.2 | コース選択画面 | 38 |
| 7.3 | 学習モード選択画面 | 39 |
| 7.4 | 通常講義モード | 40 |
| 7.5 | 教材学習時イメージ | 41 |

表 目 次

| | | |
|-----------|----------------------------|-----------|
| 4.1 | SCORM 対応プラットフォーム製品一覧（五十音順） | 13 |
| 4.2 | SCORM 対応オーサリング製品一覧（五十音順） | 13 |
| 4.3 | ナビゲーション機能実現に利用可能なメタデータの一部 | 20 |
| 4.4 | データモデルに定義される学習者情報 | 20 |
| 付録 | | 51 |
| 1 | General カテゴリ | 51 |
| 2 | Lifecycle カテゴリ | 51 |
| 3 | Metametadata カテゴリ | 52 |
| 4 | Technical カテゴリ | 52 |
| 5 | Educational カテゴリ | 53 |
| 6 | Rights カテゴリ | 53 |
| 7 | Relation カテゴリ | 53 |
| 8 | Annotation カテゴリ | 53 |
| 9 | Classification カテゴリ | 54 |

第1章 はじめに

1.1 背景

近年のインターネットの発達・普及に伴い、学習時の時間や場所の制約が比較的少なく自己学習が可能な蓄積型の遠隔学習システムが注目されている。こうしたシステムは、教育機関や企業組織などで従来行われてきた対面講義に相当する教育効果が徐々に認められてきており、講義と同等のカリキュラムを電子教材化したコースウェアを利用して正規の単位を認定するなどといった取り組みが進められつつある [1]。このような自己学習が可能な蓄積型の遠隔学習システムは、個々の学習者の意欲や理解度に合わせて学習することが可能であるため、高い学習効果が期待できる。しかしながら、従来講義室で行われてきた講義を元にコースウェアが構成されている場合には、学習者が必要とする学習項目を選択することが困難になる場合がある。例えば本学の遠隔学習システムは同様のスタイルであり、学習者が学習項目を選択時に章節構造を反映した階層的な目次を利用する必要がある。そのため、学習者がある学習項目を学習後に予習や復習の目的で次に学習する学習項目を選択する場合、異なる章や節に存在する関連した学習項目を、学習者自身が目次から探し出すことは容易ではない。こうした問題は本学システムに限定された問題ではなく、通常の講義および教科書をコースウェア化し、本学と同様の目次を学習者へと提供する遠隔学習システムでよく起こると考えられるため、このような問題の解決は蓄積型の自己学習が中心に行われる遠隔学習システムにおいて重要であるといえる。

1.2 目的

このような問題を解決する一つの方法として本研究では、「学習者にとって学習の目的にあった適切な項目を目次をとして提示する」ナビゲーション機能の実現を目指す。このナビゲーション機能を実現するために、ここでは教材情報や学習者情報を管理するための用途で一般的に利用されている学習管理システムを土台に、必要な情報を効率よく取得できるナビゲーション機能を組み込み、積極的に活用するアプローチをとる。

教材情報とは、章節構造や前提条件などの教材構造からなる情報とキーワードなどの教材の特徴を表す情報からなる。これらの教材情報はナビゲーションを行う上で地図の役割を果たすものといえる。一方、学習者情報とは学習者が過去に学習した項目に対する学習時間や、ある学習項目に対するテストの結果などといった学習履歴情報である。これら教材情報と学習者情報を重ね合わせることにより、学習者が現在学習を進める上でどのよう

な状態であるかを学習状態として表現する．さらに，ここで表現された学習者の学習状態と学習者の目的を反映して適切な学習項目を選びだすことによりナビゲーション機能を実現する．

また，現在学習管理システムにおける標準規格として SCORM が注目されている [2]．SCORM においては，XML ベースの教材構造定義やクライアント・サーバ間を流れるデータの共通化が行われており，SCORM 対応の学習管理システムであれば，教材の共有・再利用が可能となっている [2]．そこで本研究では，SCORM において定義される情報の中からナビゲーション機能の実現に向けて必要となる基盤情報を抽出・整理することによって，様々な目的のナビゲーションを効果的に実現する枠組みを提案する．さらに標準規格を用いることにより，システム自身の汎用化にも焦点を当てて議論していく．

本研究においてはナビゲーション機能を実現する上で，学習状態モジュールとナビゲーション戦略モジュールの2つのモジュールを開発する．学習状態モジュールは学習管理システムで管理される教材情報と学習者情報を利用して実装する．また，学習者の目的と学習状態に基づき適切な候補を決定する役割をナビゲーション戦略として定義し，それを実現するためのモジュールとしてナビゲーション戦略モジュールを実装する．

これらの機能を実現することにより，本研究の主題であるナビゲーション機能を有する学習管理システムを開発する．また実現したシステムの問題点を洗いだし，今後の改善の指針を示す．

1.3 本論文の構成

本論文の構成は以下の通りである。

- 第2章では、ナビゲーション機能の必要性について、本学における遠隔学習システムを例に挙げ考察する。
- 第3章では、教材情報ならびに学習者情報を管理するための学習管理システムについて述べる。
- 第4章では、学習管理システム構築に利用する SCORM 規格について述べ、実際にナビゲーション機能を実現する上で利用する教材情報、学習者情報、さらに API などについて述べる。
- 第5章では、実現するナビゲーション機能の概要を述べる。
- 第6章では、ナビゲーション機能を組み込んだ学習管理システムの具体的な内容について述べる。
- 第7章では、実現されたナビゲーション機能を有する学習管理システムを用いた学習の手順について解説する。
- 第8章では、システムの問題点、改善点に関して述べる。
- 第9章では、本研究のまとめと今後の課題について述べる。

第2章 ナビゲーション機能の必要性

2.1 本学遠隔学習システムについて

従来の教育手法では、体系的な知識の伝達を目的として、あらかじめ計画されたカリキュラムに従った講義が行われてきた。これに対して、蓄積型の個別学習が可能な遠隔学習システムでは、予習や復習などといった目的に応じて、必要な項目を学習者自身のペースで学習できることが大きな特徴となっている。ところが、実際に提供される遠隔学習システムの多くは、通常の講義または教科書をコースウェア化したものであるため、その構造は章節構造として体系化されたものになりやすい。

本章では、同様に通常講義を電子教材化しコースウェアとして学習者に提供する本学で開発された遠隔学習システムを例として挙げ、問題点についての解説および解決策について議論する。

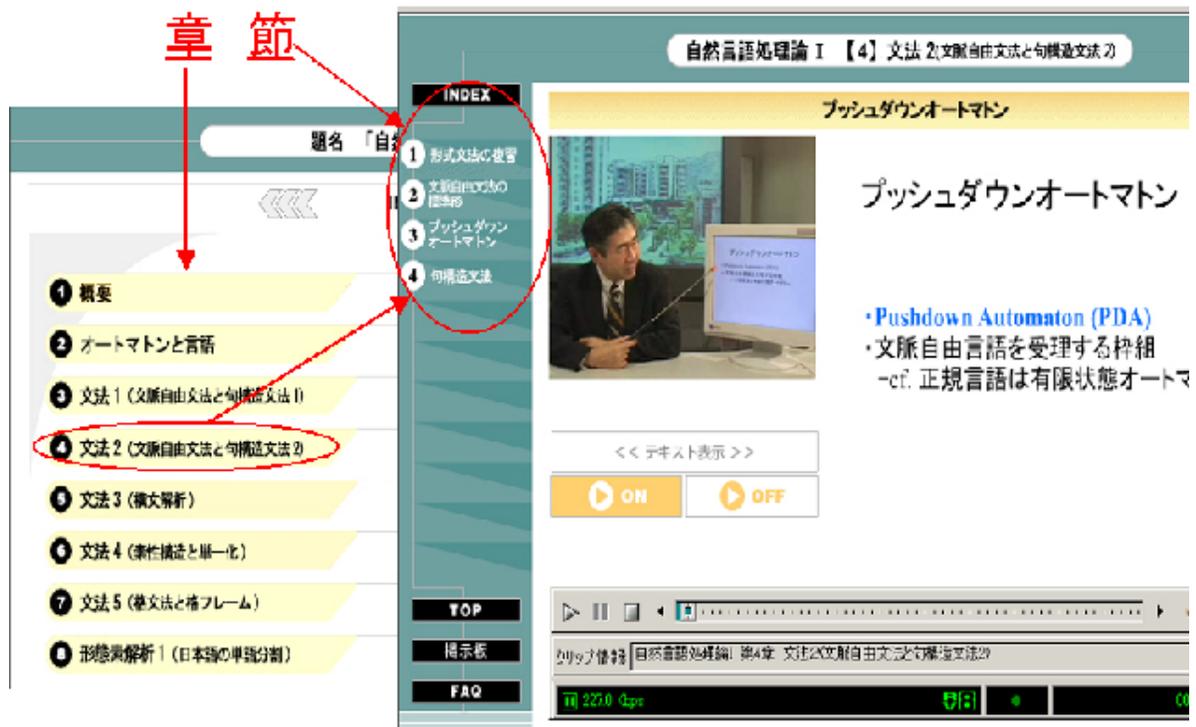


図 2.1: 本学の遠隔学習システム

本学システムは、通常講義と同等の授業内容をコースウェア化し、インターネットを利用して学習者へと作成された教材を配信する自己学習が可能な Web ベースの蓄積型遠隔学習システムである。学習者にはまず図 2.1 で示すように目次が index として提供され、学習者はその中から学習したい項目を選択し学習を開始する。さらに教材が再生された後、章よりも細かい単位である節にあたる項目が index として再生される教材の中に表示されるため、学習したい節を選択することにより、より細かい学習項目の学習が可能となる。

このような階層的な構造の目次を利用して学習者が学習項目を選択する場合、「関連する項目を学習したい」、「前提となる項目に戻りたい」などといった、章節構造以外の観点で関連した項目を学習者自身が見つけ出すことはそれほど容易ではない。

例えば、図 2.2 において、ある学習者が 8 章の 2 節にある「解析アルゴリズム」を復習目的で学習していたとする。もし学習者がここで学習している「解析アルゴリズム」に関連する学習項目を探す場合にはここで提供されるような章節構造からなる index からでは、他のどの章、どの節に関連する項目が含まれているのかを見つけ出すことは容易ではない。この例では、8 章 2 節は 5 章の 3 節から 6 節にある構文解析アルゴリズムの 1 から 4 までと「分割」「変換」という観点で関連がある。しかし、それらのキーワードで二つの章がつながっているという情報を知る術は、学習者には与えられていない。



図 2.2: index 利用時における関連項目検索の難しさ

このように学習者に対して目次である index を与え，学習を進めていく遠隔学習システムにおいて起こる問題は，本学システム固有の問題ではない．故に，このような問題を解決するようなナビゲーション機能は重要であると考えられる．

第3章 学習管理システム(LMS)

3.1 学習管理システムとは?

学習管理システム(LMS)とは、その名が示す通り「学習を管理するシステム」である。詳しく言えば、オンライン学習コンテンツや対面学習プログラムなどの学習資源の一元管理、学習者の学習履歴情報管理、人材マネジメントなどを行い、学習者とその管理者にとってより効果的な学習環境を提供するシステムである。

コンピュータによる教育支援の技術としてCAI(Computer-Assisted Instruction)[3]やCSCL(Computer-Supported Collaborative Learning)[4]と呼ばれるものが、教育の質すなわちその効果と効率の向上を目的として開発されてきた。その流れにインターネットの技術の流れが統合されたものが、WBT(WebBasedTraining)[5]と呼ばれるものであり、WBTにて教材の配布、また学習者の情報、教材の情報を効率良く管理するシステムとして用いられるのが学習管理システムである。これらネットワークにおいてWBTを用いて学習することを、現在一般的にはe-Learningと呼ぶ。e-Learningの定義は人によって異なるが、以下の点は共通しているといえる。

- ネットワークやインターネット技術を活用し、コンピュータを介して瞬時に更新、検索、配布、共有を行う
- 知識やパフォーマンスを向上させるために、伝統的な訓練(トレーニング)の枠を越えた広範囲な解決法を提供する学習ソリューションである

それでは学習管理システムの代表的な機能として挙げられる、教材配信、コミュニケーション、成績進捗管理、スキル管理、カリキュラム管理、利用者インタフェース、外部APIについて解説する[2]。

- 教材配信
教材配信機能は、教材コンテンツを学習者端末に提示し、学習履歴を学習ログとして保存する。いわゆる狭義のWBTシステムはこの機能を有する。通常のWWWによるマルチメディアデータの配信と異なるのは、最低限学習者ごとのログが保存、管理される、いわゆるトラッキング機能を有する点である。
- 教材作成
教材作成機能は教材コンテンツを作成する機能で、コンテンツの種別に応じて種々

のツールを使用する．一般的に言われるオーサリングツールであり，教授者，ならびに教材作成者である業者などにより利用される．

- コミュニケーション
コミュニケーション機能とは，生徒対教師や生徒同士のグループ学習を実現するためのものである．メール，掲示板などの非同期型のツール，チャット，アプリケーション共有などの同期型ツールがこれに相当する．
- 成績進捗管理
成績進捗管理機能は，学習進捗状況や成績管理などの学習履歴を学習者自身や管理者が把握するために用いる．教育研修の効果把握や，学習者ならびに教授者が学習を進める上で必要不可欠な機能である．
- スキル管理
スキル管理は，職種と知識，スキルと研修カリキュラムのような体系を管理し，個々の学習者に知識，スキルレベルと照らし合わせて個人ごとの研修カリキュラムを提示するための機能である．管理者側からみると，ある職種に必要な人材が企業内に何人いて，企業の目標達成のためにどのような教育研修を実施すべきかを把握するという，企業戦略上重要な機能である．
- カリキュラム管理
カリキュラム管理は，コース間の依存であるとか，期間や受講人数の制限などカリキュラム内，カリキュラム間の教育研修的な，あるいは物理的な制約条件を管理し，スムーズな進捗を支援する．
- 利用者インタフェース
利用者インタフェースは，学習者に提示される GUI やメニューである．個人情報や履歴に基づいて情報を提示するようなシステムもある．
- 外部 API
外部 API は，学習管理システムが外部と連携するためのインタフェースである．学習管理システムで扱うデータである各種情報にアクセスするための関数メソッドを提供する．

さらに様々な機能も考えられるが，各々開発する学習管理システムごとに固有の機能もあるので，この機能を持ってこそ学習管理システムである，という定義は本研究では行わない．

次に，一般的な学習管理システムのイメージを図 3.1 で示す．

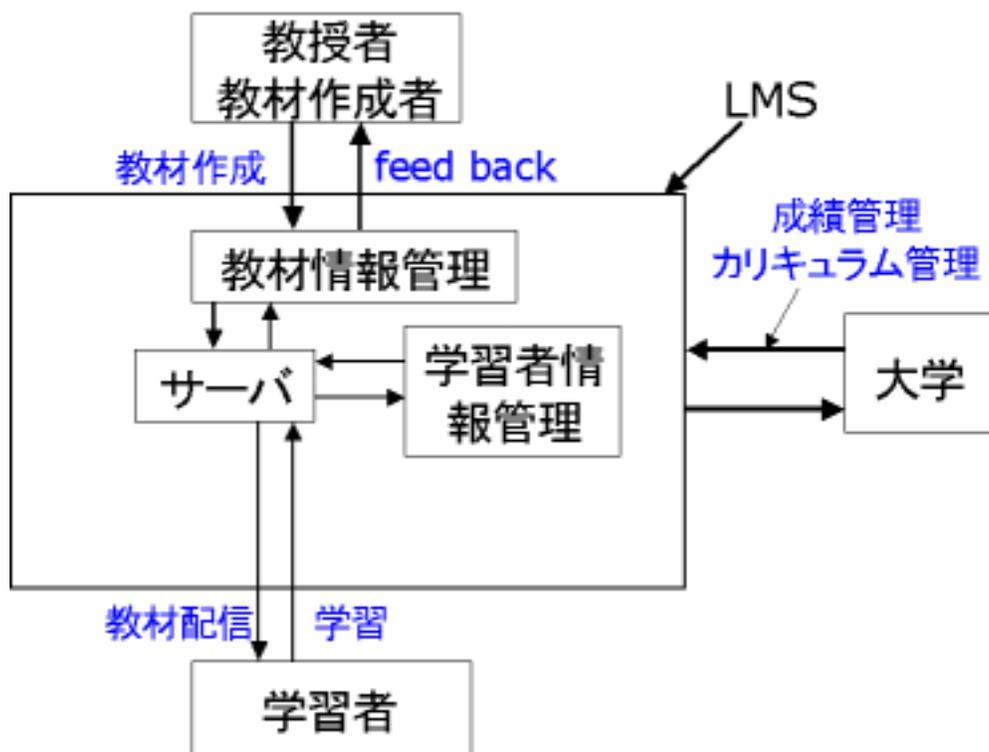


図 3.1: 従来の学習管理システムのイメージ

図 3.1 で示す学習管理システムには，まず学習者に教材を配信するための教材情報を管理する教材情報管理機能，次に学習時間やテストの得点などの学習者履歴情報を管理する学習者情報管理機能がある．また，教授者ならびに教材作成者は学習者からのフィードバック情報を利用し，教材の修正をしたり，オーサリング機能を利用して新しい教材を作成し，教材情報管理機能へと登録する．さらに，学習管理システムは大学などと連携し，学習者の成績管理であるとか，カリキュラム管理などを行う場合もある．

本研究で実現するのはナビゲーション機能を有する学習管理システムである．そのナビゲーション機能を開発する上で，学習者の状態を推定するために学習者情報と教材情報を利用する．それらの学習者情報，教材情報を利用して学習者に教材を配信するナビゲーション機能を有する学習管理システムのイメージを図 3.2 で示す．

学習管理システムを構築する上で標準化という話は無視できない 1 つの大きなポイントである．次節では，標準規格について説明する．

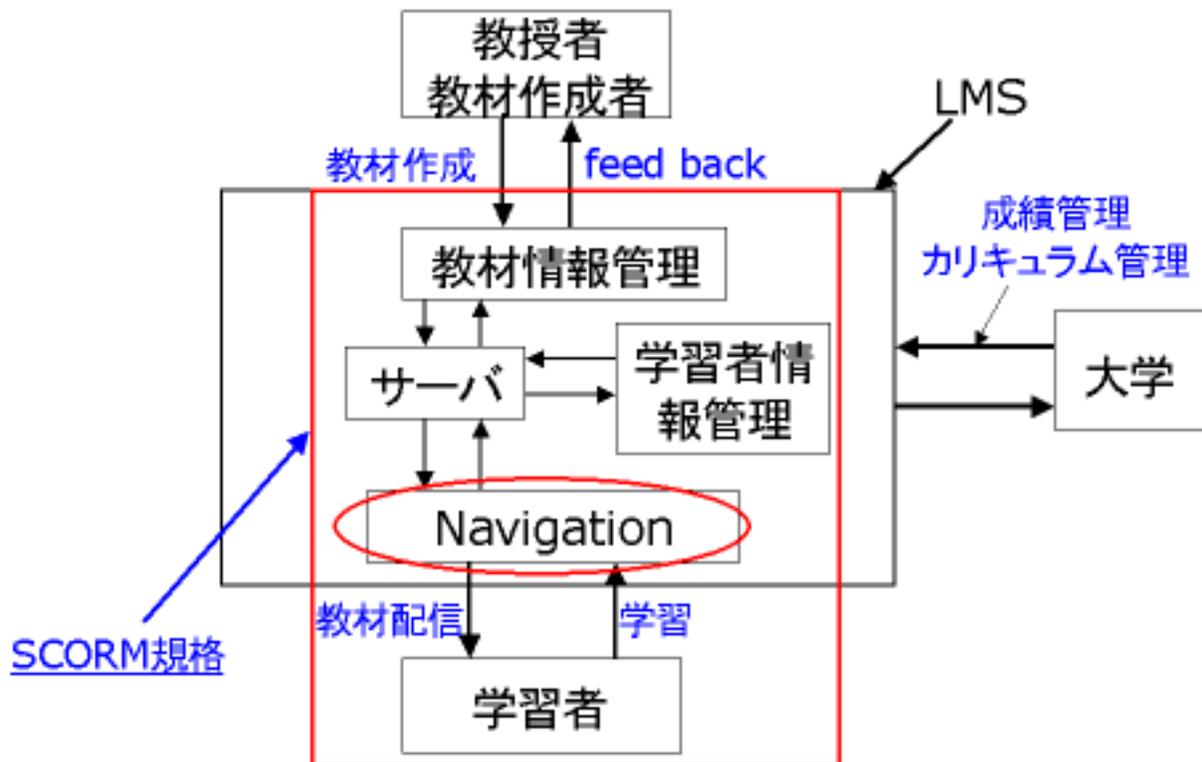


図 3.2: 本研究で開発する学習管理システムのイメージ

3.2 学習管理システムにおける標準規格

学習者に提供されるコンテンツを、どのような形であれ作成することは可能である。しかし、あるシステムで作成したコンテンツが他のシステムで再生可能であるかと言えば、教材を作成する時点で共通の取り組みを行わない限りは不可能である。つまり共通の教材作成フォーマットを利用して教材が作成されれば、異なる環境においても、教材の再生が可能である。このような教材コンテンツの標準規格が、従来からいくつか定義されている。その中でメジャーな規格である、AICC、IEEE LTSC(LOM)、IME、SCORMについて解説する。

- AICC(Aviation Industry CBT Committee)[6]
AICCとは、標準化団体「航空産業 CBT 委員会」の団体名から取って「AICC」と名づけられた。この規格では、学習管理システムとコンテンツの相互運用性と互換性を保つための統一仕様が策定されており、WBTのコース管理を最適化し、すべての教材があらゆる学習管理システム上で動作することを推奨している。すでに現時点で、この規格は事実上の業界標準としてベンダーに広く採用され、AICC準拠と銘打った製品も数多く提供されている。

- IEEE LTSC Learning Object Metadata(LOM)[7]
IEEE(Institute of Electrical and Electronics Engineers) とは，米国電気電子技術者協会のことであり，その中でも教育に関する標準化の検討は LTSC(Learning Technology Standards Committee)[8] で行われている． IEEE LTSC LOM 規格の標準化の目的は，例えば，学習者，インストラクタまたは自動化されたソフトウェアプロセスによる学習オブジェクトの検索，評価，獲得，および使用を促進することである．また，学習オブジェクトやそのメタデータが利用される教養・言語上の文脈の多様性を考慮に入れたカタログやインベントリの作成を可能にすることにより，学習オブジェクトの共通利用を促進することである．
- IMS(IMS Global Learning Consortium)[9]
IMS は政府機関，教育機関，コンピュータ・ベンダーなどで構成する非営利組織であり，分散型 Learning に関わるオープンな技術仕様の開発を目的に，1997 年に設立された．マルチメディアコンテンツ，ソフトウェアツール，教育機関など学習オブジェクトを管理・検索・関連付け・評価するための LOM 規格を開発している．
- SCORM(Sharable Content Object Reference Model)[10]
SCORM 規格は 1997 年に設立された アメリカ政府が進める標準化団体 ADL(Advanced Distributed Learning Initiative)[11] が策定した学習管理システムと学習コンテンツの相互運用性を高めるための仕様である．現在多くの遠隔学習システムは広く普及した Web 技術を元に作られているが，一般の Web サーバにはない次のような機能をサポートすることが特徴である．
 - － 学習コンテンツ全体の構造，すなわち教材の章立てにしたがって学習者にコンテンツを表示する順番を決定する機能．
 - － 学習者がどの部分まで学習したか，演習問題の解答内容，得点がどうだったか等，学習履歴の記録を行う機能．
 - － 先行して策定された AICC 規格の成果を取り入れつつ，異なった学習管理システム間で学習コンテンツの相互運用性を向上するための標準化事項．

本研究では，汎用化をも視野に入れ学習管理システムの開発を目指している．また，ナビゲーション機能を実現するうえで有用な情報として利用できる標準化規格として SCORM を利用することとする．

次章では，ここで紹介した SCORM の概要について掘り下げて解説を行う．

第4章 SCORM

4.1 概要

様々な分野において技術の標準化は非常に重要なポイントとなっているが、学習管理システムにおいてもここ1,2年の間に、SCORMと呼ばれる規格が普及しつつある。SCORMは、2000年1月にADLから提供が開始された学習管理システムの標準規格である。SCORMとはSharable Content Object Reference Modelの略であり、これまでに提案されてきたAICCのCMI規格やIMSのLOM規格などといった様々な標準規格を集約したものとなっている。この規格に基づいて学習管理システムを構築することにより、SCORMに準拠する教材コンテンツの相互運用と再利用が可能となる。

具体的な仕組みとしては、XMLを用いたコンテンツ集約モデルによる教材へのメタデータの付加と、階層型の教材構造の定義に加えて、クライアント・サーバ間の通信用APIの定義とクライアント・サーバ間の通信時に流れるデータの共通化があげられる。ただし、SCORMはあくまでも標準規格であり学習管理システムそのものではないため、教材コンテンツを配信する学習管理システム自身は各自構築する必要がある。

表4.1にSCORM対応の学習管理システムの製品一覧を、表4.1にSCORM対応のオーサリングツールの製品一覧を示す[12]。

表 4.1: SCORM 対応プラットフォーム製品一覧 (五十音順)

| | | |
|----|-----------------------|--------------------------------|
| 1 | (株) エヌ・ティ・ティ・ エックス | Xcalat |
| 2 | NTTラーニングシステムズ(株) | Learning Station V7.0 |
| 3 | コンパック(株) | AcademicWare WBT |
| 4 | (株) スマートリンク | Pratica SCORM Adapter |
| 5 | トランス・コスモス(株) | TEASY R2 |
| 6 | 日本電気(株) | Cultiiva Enterprise |
| 7 | 日本ユニシス(株) | Virtual Campas |
| 8 | (株) ネットラーニング | NetLearning Platform |
| 9 | 日立電子サービス(株) | HIPLUS |
| 10 | (株) 富士通インフォソフトテクノロジー | Internet Navigware Server V7.0 |
| 11 | 三菱電機インフォメーションシステムズ(株) | 三菱 e-Learning/HRMS |
| 12 | 横河インフォテック(株) | ActiWise |
| 13 | ロータス(株) | ロータスラーニングスペース 5.0 |

表 4.2: SCORM 対応オーサリング製品一覧 (五十音順)

| | | |
|----|-----------------------|--|
| 1 | (株) エヌ・ティ・ティ・ エックス | Xcalat Author |
| 2 | NTTラーニングシステムズ(株) | Learning Station 教材作成キット V7.0 |
| 3 | (株) オール・アンド・エニー | ToolBook Instructor8.1 |
| 4 | コンパック(株) | AcademicWare Author |
| 5 | (株) スマートリンク | Pratica SCORM Processor/Editor |
| 6 | (株) デジタルナレッジ | InstructNow, V3.0.SeminarNow, V3.0.RemindNow |
| 7 | (株) ネットラーニング | NetLearning Platform |
| 8 | 日立電子サービス(株) | HIPLUS Author |
| 9 | (株) 富士通インフォメソフトテクノロジー | Internet Navigware 教材作成キット V7.0 |
| 10 | 横河インフォテック(株) | ActiWise |
| 11 | (株) レイル | LiveCreator ver3.0 |

4.2 コンテンツ集約モデル

SCORM では、教材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として、コンテンツ集約モデル [13] が定義される。ここでいう学習資源は、学習体験で利用されるあらゆる表現であり、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。学習体験を生成、提供するプロセスにおける過程の一つとして、単純な学習オブジェクトを生成、検索し、これをさらに複雑な学習資源に集約し、あらかじめ決められた順序で配布する過程がある。SCORM で定義されるコンテンツ集約モデルはこの過程を支援するもので、以下のように構成される。

- コンテンツモデル
学習体験におけるコンテンツ構成要素を規定する用語
- メタデータ
コンテンツモデル構成要素の特定のインスタンスの属性や特徴の記述を行うための仕組み
- コンテンツパッケージング
学習体験の意図したふるまいの記述方法（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）

次節より、これらコンテンツ集約モデルに含まれる 3 つの要素を解説する。

4.3 コンテンツモデル

SCORM において教材は、Asset、SCO、コンテンツアグリゲーションの 3 種類の概念で表現される。Asset とは HTML ファイルや動画ファイル、画像ファイルなどの単体の学習オブジェクトのことであり、SCO はそれら Asset が複数集って、学習管理システムとの通信機能を持った学習項目である。

さらに図 4.1 で示すように、複数の SCO を集約したものがコンテンツアグリゲーションであり、SCO を集約することにより、コースや章、節などといった教材構造を表現することができる。SCORM 対応の学習管理システムは XML ファイルで定義された階層型の教材構造を参照することにより、次にどの項目を学習者に提示するのかを決定することが可能となっている。

こうした、Asset、SCO、コンテンツアグリゲーションなどといった教材の構成要素に関する命名規則をコンテンツモデルという。

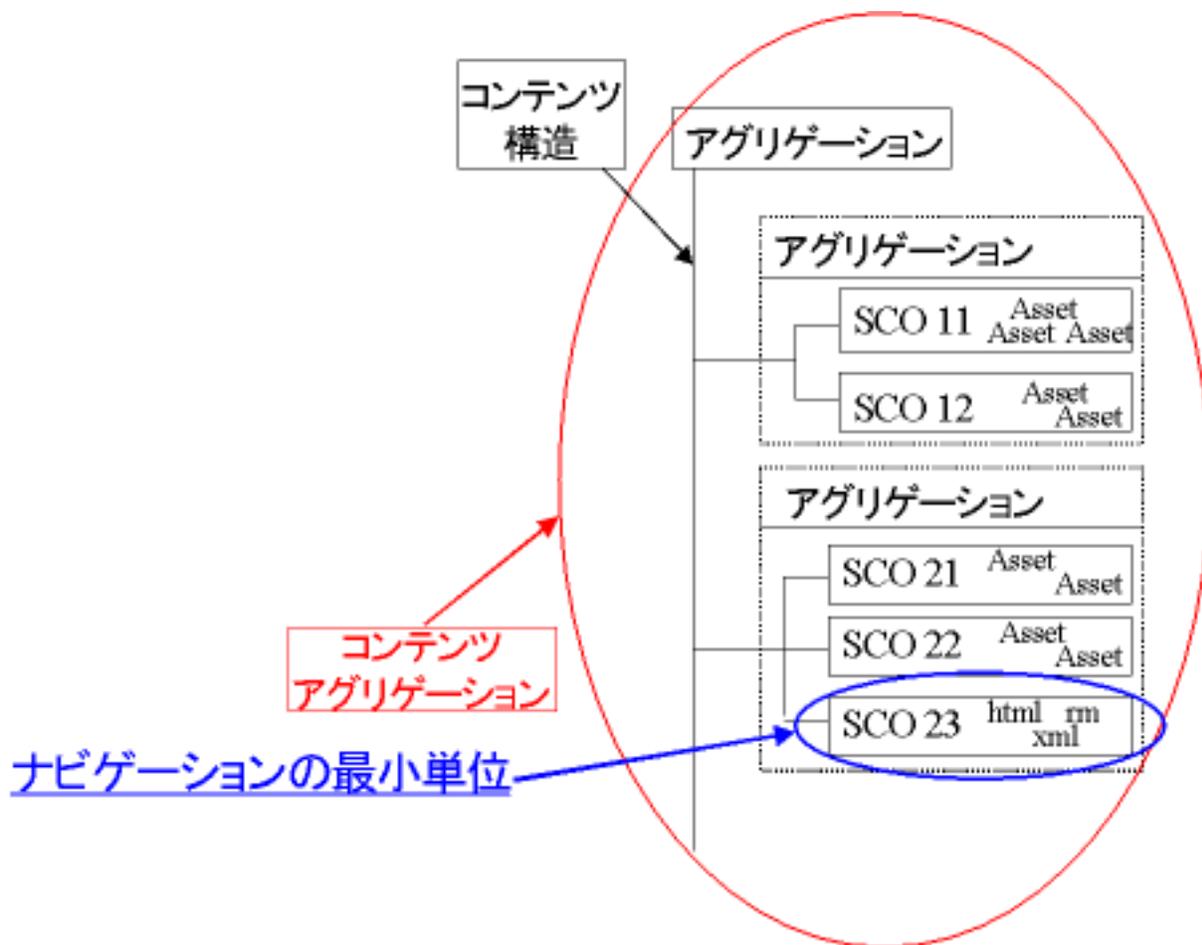


図 4.1: コンテンツアグリゲーション

4.3.1 Asset

学習コンテンツの最も単純な形態は、Asset で構成される。Asset は他の Web クライアントに配布可能な、メディア、テキスト、画像、音声、Web ページ、アクセスオブジェクトの電子的な表現やデータである。Asset の属性や特徴は Asset メタデータで記述され、これによってオンラインリポジトリの中での検索・発見が可能となり、再利用の機会を向上させることができる。

4.3.2 SCO

SCO(Sharable Content Object) は、SCORM ランタイム環境を利用して学習管理システムと通信を行う学習ユニットであり、複数の Asset の集合である。SCO は SCORM ランタイム環境によって学習管理システムが動作を記録できる最小単位の学習資源である。

再利用可能であるためには、SCO は学習文脈から独立してはならない。例えば SCO は異なる学習経験において異なる学習目標に使用できる。さらに、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。

複数の学習目的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の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、実行時に学習管理システムによって動作を記録したいコンテンツの最小の理論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は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 メタデータで記述され、これによってオンラインリポジトリの中での検索・発見が可能となり、再利用の機会を向上することができる。

4.3.3 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、学習資源を一つの結合した学習ユニット（すなわち、コース、章、モジュールなど）に集約し、コンテンツ構造を定義することである。コンテンツ構造を定義することにより、学習資源をユーザに提示する順番を定義する機能が提供される。

4.4 メタデータ

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。SCORM で扱うメタデータは、IEEE LTSC 学習オブジェクトメタデータ (LOM) 規格を基礎として、IMS と ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe)[14] 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。また、SCORM は、IMS 学習資源メタデータ情報モデルのメタデータ要素と同じセットを採用し、さらに IMS 学習資源メタデータ XML バインディング仕様をも採用している。SCORM は、IMS メタデータ要素の定義を 3 つのコンテンツモデルの構成要素である、Asset、SCO、コンテンツアグリゲーションに適用している。これら 3 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。

SCORM で定義されるメタデータ情報は 9 つのカテゴリに分けられる。以下にそのカテゴリを示す。

1. 「一般 (General)」: 資源の属性、特徴を統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 (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. 「他のオブジェクトとの関連 (Relation)」: この資源と他のオブジェクトとの関係付けの定義
8. 「注釈 (Annotation)」: 教育的使用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

さらにこれらのカテゴリの下に, 複数のメタデータ要素が定義されている. 実際に定義される全てのメタデータは付録にて示す.

4.5 コンテンツパッケージング

コンテンツパッケージングの目的は, 異なるシステムやツールの中でデジタル学習資源を交換する標準化された方法を提供することである. コンテンツパッケージングは, 学習資源の集合の構造と教材作成者が意図する動作を以下のように定義することができる.

- パッケージそのものについての記述, および以下を包括するマニフェストファイル
 - パッケージに関するメタデータ
 - コンテンツ構造と動作を定義する任意のオーガニゼーションセクション
 - パッケージが入っている資源の参照リスト
- XML ベースのマニフェストの作成方法
- マニフェストとすべての関連する物理ファイルを zip ファイルないし, CD-ROM などにパッケージングするための指示

コンテンツパッケージングは, デジタル学習資源やその集合を学習管理システム, 開発ツール, コンテンツリポジトリ間でインポート, エクスポートするために使用することを想定している. コンテンツパッケージング仕様はどのようなシステムでもサポート可能な共通の「入出力」フォーマットである.

4.6 データモデルとAPI

SCORM 対応の学習管理システムと学習者に提供される SCO は通信を行うことが可能である。この通信においてやりとりされる情報は、学習者の氏名やID、そのSCOに対する学習者の学習時間などである。これらの学習管理システムとSCOの間に流れるデータはデータモデルとして定義されている [15]。

図 4.2 に示すように、実際にSCOはクライアントに組み込まれるAPI Adapterを経由して学習管理システムと通信する。この間を流れるデータを共通化することにより、異なるSCORM対応の学習管理システムにおいても相互運用が可能となる。

4.6.1 データモデル

データモデル (Data Model) は、学習者の状態などといった伝達される情報を定義するために使われるデータ要素の標準セットである。こうしたのデータモデルを確立する目的は、異なる学習管理システム環境によっても、SCOに関する定義された情報セットを確実に追跡することができるようにすることである。例えば、学習者の得点追跡が一般的な要求であることとした場合、SCOは学習管理システムに得点を報告するための共通の方法を確立する必要がある。なぜなら、もしSCOが特有の得点表現を使用するならば、学習管理システムはその情報をどのように受け取り、格納し、処理する方法を理解することができないといったことが起きる。

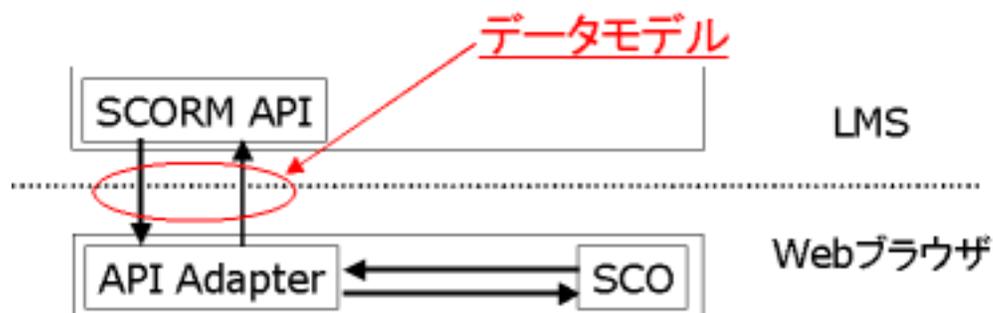


図 4.2: データモデル

4.6.2 API

共通のAPIを使用することにより、相互運用性と再利用に関する多くのSCORMの要求を満たすことができる。共通のAPIは、SCOが学習管理システムと通信するための標準化された方法を提供し、同時に、コンテンツ開発者に対して通信機能の個別の実装を隠蔽することとなる。最も単純な言い方をすれば、APIは単なる限定の機能セットであ

り、SCO は利用可能なものとしてそれに依存することができる。API アダプタは API の機能を実現・提示する機能ソフトウェア部品である。コンテンツ開発者が同じ公開インタフェースを使用する限りは、API アダプタの内部実現方法は、コンテンツ開発者が意識すべき事項であってはならない。学習管理システムは単に、API の機能を実現し、クライアント SCO にそのインタフェースを提示する API アダプタを提供する必要がある。

API の重要な面は、SCO が学習管理システムと通信することを可能にするコミュニケーション・メカニズムであるということである。ひとたび SCO が起動されたならば、SCO は学習管理システムと情報を”get”したり”set”したりする。API アダプタと SCO 間の全てのコミュニケーションは SCO によって開始される。

4.7 ナビゲーション機能に必要な情報

SCORM に対応した学習管理システムでは、様々な教材情報や学習者情報が規格として定義されている。ここでは、ナビゲーション機能を実現するために利用可能な情報について整理する。

4.7.1 教材情報

SCORM においては各 SCO に対して、メタデータを定義した XML ファイルが付加されているだけでなく、教材構造も XML にて定義されているので、それらを参照することにより、教材情報を獲得することができる。定義されるメタデータから、ナビゲーション機能にて利用可能なものを挙げ、表 4.7.1 にて表す。これらの情報を利用することで、教材の標準再生時間や、難易度、キーワードなどといった情報を参照することができる。

4.7.2 学習者情報

SCORM のデータモデルの中には、いくつかの学習者情報が定義されている。表 4.7.2 に示すように、SCO に費やした学習合計時間を表す `total_time` や、対象の SCO は学習したものなのか、初めて学習するものなのかを表す `lesson_status`、通常講義、予習、復習のどのモードで SCO を起動したかについて表す `lesson_mode` などが定義されている。データモデルで定義されるこれらの情報は随時クライアントから取得することができる。また、これらの情報は学習履歴情報としてデータベースに格納されている。

表 4.3: ナビゲーション機能実現に利用可能なメタデータの一部

| 各要素 | 各要素に関する情報 |
|---------------------|-------------------------|
| title | 学習資源に与えられた名前 |
| language | 学習資源をやりとりする時に使用する言語 |
| keyword | 特徴を表すキーワードやフレーズ |
| aggregationlevel | 学習資源の機能的な粒度 |
| version | 資源のバージョン |
| size | デジタル資源のサイズ |
| location | 資源の格納場所を表す文字列 |
| requirement | 資源を利用する場合の技術的な要件 |
| duration | 資源を想定された速度で再生した場合に要する時間 |
| typicalagerange | 想定されるユーザの年齢層 |
| difficulty | 難易度 |
| typicallearningtime | 標準学習時間 |
| kind | 対象資源との関係 |
| resource | 対象となる資源 |

表 4.4: データモデルに定義される学習者情報

| | |
|--------------------------|---|
| cmi.core.student_id | 学習者を識別するユニークな ID |
| cmi.core.student_name | コース名義上の学習者名 |
| cmi.core.lesson_location | 学習者が学習した SCO の出口ポイント 再学習にエントリポイントとして利用 |
| cmi.core.credit | 学習者の履歴記録が学習管理システムによって更新されるか |
| cmi.core.lesson_status | LSM によって決定される 6 つのステータス値 passed, completed, failed, incomplete, browsed, not attempted |
| cmi.core.entry | 学習者が SCO に以前に入ったことがあるかどうか |
| cmi.core.score | 学習者の得点表示 |
| cmi.core.score.row | SCO に対しての最終試行における学習者の得点 |
| cmi.core.score.max | 最高点 |
| cmi.core.score.min | 最低点 |
| cmi.core.total_time | SCO のトータル学習時間 |
| cmi.core.lesson_mode | SCO の起動モード (browse, normal, review) |
| cmi.core.exit | 学習をどのようにして終了したか |
| cmi.core.session_time | 学習を終了するまでに費やした合計時間 |

第5章 ナビゲーション機能の実現

本研究で実現するナビゲーション機能は、以下の2つのモジュールを用いることで実現する。

- 学習状態モジュール
- ナビゲーション戦略モジュール

これにより、学習状態モジュールが候補となる学習項目である SCO 群を挙げ、上位モジュールにあたるナビゲーション戦略モジュールにおいて、それら候補の中から最適な項目を選択し、学習者へと候補となった学習項目を強調表示するような index を作成して提示する。本章では、それら2つのモジュールについて解説する。

5.1 学習状態モジュール

学習者の状態を表現するモジュールが学習状態モジュールである。本研究では、複数の観点から学習状態を切り出し利用する。なぜこのような複数の観点からなる学習状態を考えるかという点、1つは学習者の様々な目的に適用可能な多様なナビゲーションを想定しているからである。次に、複数の学習状態を利用することにより、新しい要求が発生したときに、新たにそれに対応する学習状態が追加できるような拡張性について考慮することができるからである。さらに考慮する点は汎用性である。あるシステムにおいて、例えば教材にテストがなく、得点という学習者の履歴情報を持っていないとする。その場合に、学習状態が複数利用することが可能であれば、得点を利用する学習状態を用いるナビゲーションを利用しなくとも、他の情報(例えば、学習時間など)を使ったナビゲーションを利用することができる。このように本学の環境だけでなく、他の学習管理システムの環境をも考慮することにより、SCORM で実現される汎用性だけでなく、システム自体の汎用化も促進できる。

例として以下の3つの学習状態を挙げ、ナビゲーション機能を説明する。

1. 章節構造および得点
2. 前提条件および学習時間
3. キーワード関連および学習時間



図 5.1: 学習状態モジュール

例として1. について解説する。

1. は「章節構造および得点」を用いて表現される学習状態である。ある学習者が現在図 5.1 に示すように SCO22 を学習しているものとする。その時に、教材構造の情報である章節構造より、SCO22 の前に、SCO21 が、後に SCO23 があることがわかる。またこれら SCO21, SCO23 は以前に受けたテストの点数としてそれぞれ 80 点, 60 点という学習者履歴情報を持っている。この場合に、これら SCO21(80 点), SCO23(60 点) という情報を持つ 2 つの SCO が「章節構造および得点」からなる学習状態より候補として選びだされる。

このようにそれぞれの各学習状態から候補となる SCO を選びだすと、図 5.2 に示すように学習状態モジュール群が構成され、候補となる SCO 群がそれぞれの学習状態より挙がることとなる。

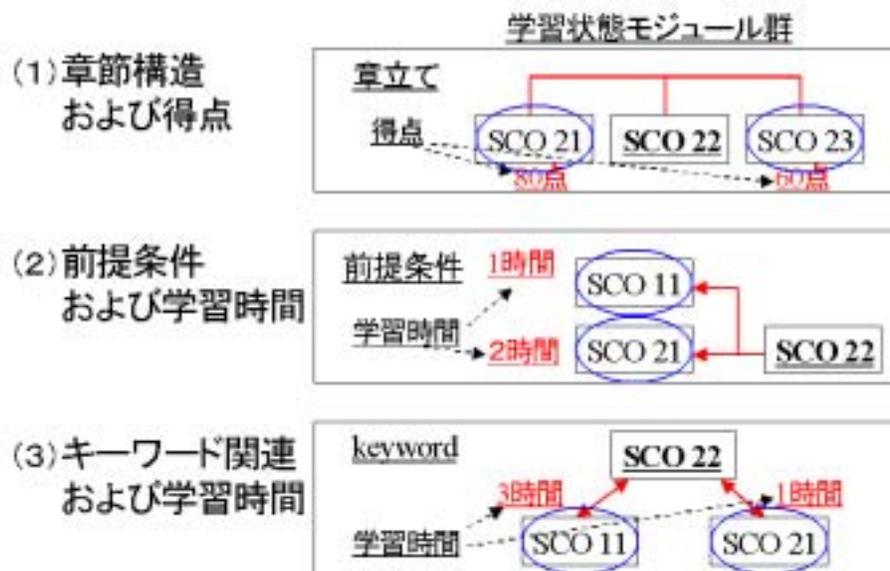


図 5.2: 学習状態モジュール群

次にここで候補として挙げた SCO 群を扱い、実際に最適な学習項目を選びだし学習者へと提供する index を作成するナビゲーション戦略モジュールについて説明する。

5.2 ナビゲーション戦略モジュール

ナビゲーション戦略モジュールは、図 5.3 で示すようにまず学習状態モジュール群で候補として挙げられた SCO 群を取得する。ここでは

- 「章節構造および得点」より SCO21(80 点) と SCO23(60 点) が
- 「前提条件および学習時間」より SCO11(1 時間) と SCO21(2 時間) が
- 「キーワードおよび学習時間」より SCO11(3 時間) と SCO21(1 時間) が

それぞれ候補としてナビゲーション戦略モジュールへと渡される。

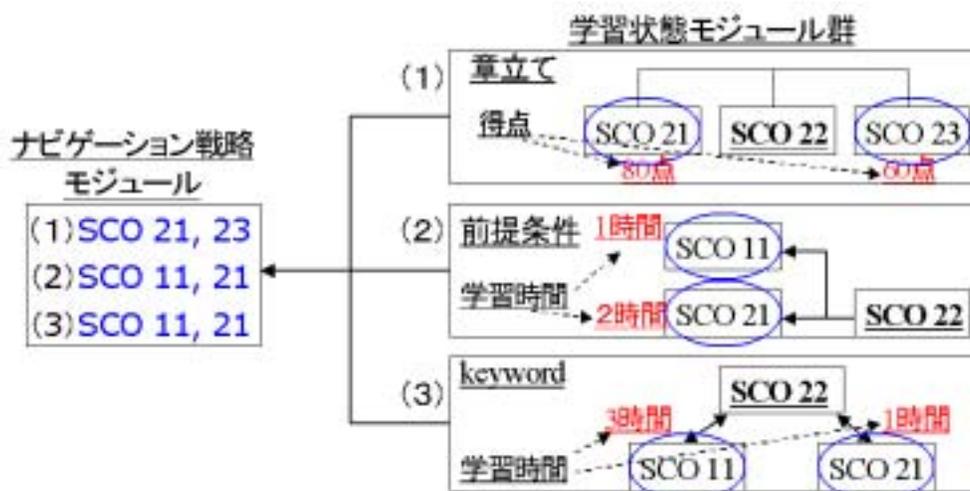


図 5.3: ナビゲーション戦略モジュールと学習状態モジュール群

ナビゲーション戦略モジュールは、ここで学習状態モジュール群より取得した SCO 群を単純に集計して学習者へと提示するわけではない。ここで学習者の目的に合うような適切な学習項目を選びだす必要がある。ここでは大きく 2 つの重み付けを行う。

1. 予習，復習，通常講義による学習目的に対する重み付け
2. 個人の学習パターンに基づく動的な重み付け

1. に関しては、本研究では学習者の学習目的を大きく、通常講義，予習，復習の 3 つに分類している。通常講義モードであれば次へ次へと学習の進む章節構造が学習を行う上で

重要であると考えられるため、章節構造で次の項目が優先されることとなる。予習であれば、自分が学習した学習項目と前提条件やキーワードなどで関連するまだ学習していない学習項目に対して重み付けが行われる。復習に関しては、各学習状態より関連する学習状態モジュールが候補となり、2.で行う個人の学習方法に基づく動的な重み付けが重要となる。

次に2.の個人の学習パターンに基づく動的な重み付けについて考える。1.で示した大まかな学習目標にて行う重み付けだけでは、本研究で定義した「学習者にとって学習の目的にあった最適な項目をindexとして提示すること」というナビゲーションを支援することは難しい。なぜかと言えば、学習者それぞれが大きな目的として、予習、復習、通常講義という学習目的を持っていたとしても、各学習者の学習方法は、そこから先大きく異なるからである。例として、学習状態モジュール群により取得された候補SCOをどのようにして学習者の目的にあった候補として選びだすかを示す。

まず、学習者が取得した候補SCO群を単純に集計した例を図5.4に示す。

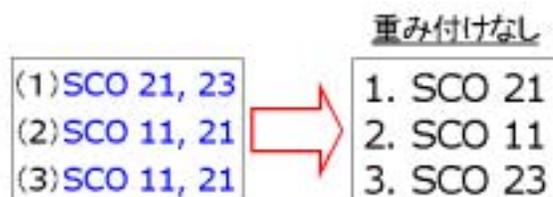


図 5.4: ナビゲーション戦略モジュール1

ここではまず、学習状態モジュール群から取得した候補SCO群を単純集計し、数の多いものから学習者へと提示している。ここで示す(1),(2),(3)はそれぞれ

- (1) 章節構造および得点
- (2) 前提条件および学習時間
- (3) キーワード関連および学習時間

からなる学習状態を表す。

1番目の候補として、SCO21が挙がり、2番目にはSCO11が、3番目にSCO23が候補として挙がっている。ここで候補として挙がる順番に基づき学習者へと強調表示されたindexが提供されたとする。学習者はそれら強調された候補が含まれたindexの中から次に学習する学習項目を選択することとなる。図5.5で示すように、ここでは学習者は候補として3番目に挙げられたSCO23を選択したとする。

学習者が選択したSCO23は(1)で示す「章節構造および得点」からなる学習状態からのみ挙がった候補である。SCO23はさらに過去のテストの得点が60点であったという情



図 5.5: ナビゲーション戦略モジュール 2

報も持っている。このことで次も「章節構造および得点」の中でも「得点の低い項目」を選択することが予想される。

学習者はそこで選択した SCO23 を学習し始める。次に学習者が SCO23 を学習し終わった時に、システムは SCO23 の後に候補として提示する SCO を決定しなければならない。最初の場合と同様に、3 つの学習状態である (1), (2), (3) より、候補となる SCO 群は図 5.6 で示すように挙げられる。



図 5.6: ナビゲーション戦略モジュール 3

ここで挙げた SCO 候補を、先ほどと同様に集計すると 1 番目の候補として、SCO12 が、後は同順の 4 番目の候補として、SCO21, 22, 24 という順番に強調されて学習者へ index として提示される。

ここで、「章節構造および得点」の「得点の低い項目」を優先させることによって、学習者に提示する SCO の候補というものは図 5.7 で示されるような形となる。

図 5.7 で示されるように、重み付けなしで集計した場合には、SCO12 が第 1 候補として提示されていたのに対し、(1) を重み付けすることにより、「章節構造および得点」からなる学習状態が重視され、さらに「得点の低い項目」が学習者に第 1 番目の候補として提示



図 5.7: ナビゲーション戦略モジュール 4

されることになる。ここで学習者が第 1 番目の候補として提示された SCO24 を選択すれば、ここで行ったナビゲーション機能は有効であるといえる。

このように各学習者の学習の進め方に対して重み付けを行うことにより、個々の学習者の学習の目的にそった重み付けを行うことが可能となり、「学習者にとって学習の目的にあった最適な項目を index として提示すること」というナビゲーション機能が実現する。

5.3 ナビゲーション機能

本研究で実現する、学習状態モジュール群とナビゲーション戦略モジュールを合わせたナビゲーション機能のイメージを図 5.8 に示す。

まず学習状態モジュール群であるが、それぞれ共通の呼び出し方法が定義され組み込まれる形となっている。それにより上位モジュールであるナビゲーション戦略モジュールから、同様の呼び出し方法で学習状態モジュール群を呼び出すことができる。

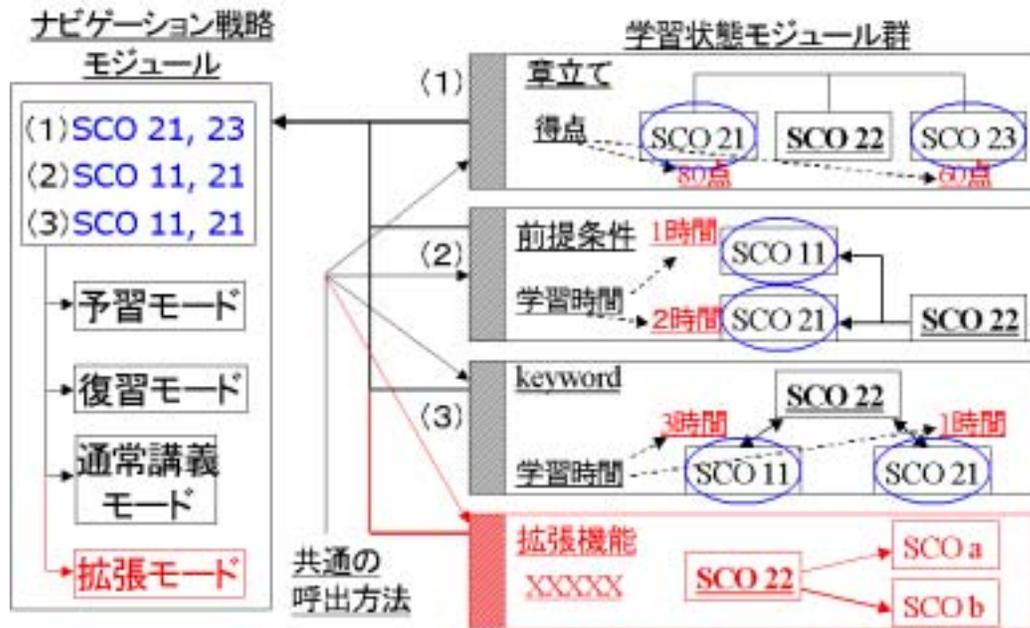


図 5.8: ナビゲーション機能

以下に実際に学習状態モジュール群に組み込む共通の呼び出し方法である Java 言語で定義した interface のプログラムを示す [16] .

```
package lmscmi.navi;

import java.io.*;
import java.util.*;

public interface Get_Sco_IF {
    public abstract void set_Sco( String sco );
    public abstract Vector get_Sco();
}

```

このプログラムはそれぞれの学習状態モジュールへと組み込まれ, abstract 宣言がされている関数 set_Sco と get_Sco の 2 つを, 各々の学習状態モジュールで実装する必要がある. set_Sco 関数は引数として学習し終わった SCO を値としてモジュールへと渡し, 各モジュールは必要な処理を行い, 候補となる SCO を選びだす. そこで候補となった SCO 群を上位モジュールであるナビゲーション戦略モジュールが, set_Sco 関数を呼び出し値を取得する.

このように Java 言語の interface を利用することにより, 将来的に追加されるであろう学習状態モジュールに対しても同様の値の渡し方と呼び出し方法が保証され, 拡張性とい

う面でも効果的なシステムを構築することができる。

次に、ナビゲーション戦略モジュールについて説明する。ナビゲーション戦略モジュールの処理は学習目的に合わせて大きく予習モード、復習モード、通常講義モードの3つに分類している。それら大きな学習目的によって、学習者へと提示する index を作成する場合に、基礎となる重み付けが変化する。そして各学習者の学習の進め方に対しても重み付けを行うことにより、基礎となった重みが各学習者の学習パターンに基づいて動的に変化する。さらに、単純に学習者が選ぶこれら3つのモードだけではなく、拡張機能として教授者が学習者に対して優先させて学習を行わせたいモードなどを追加することができる。例えば、学習者に対して index を提示せずに単純に次へ、次へと章節構造に沿って学習させたいモードなどの追加が考えられる。

このように、学習状態モジュール群とナビゲーション戦略モジュールを組み合わせ本研究で行うナビゲーション機能を実現した。

第6章 システムの概要

6.1 システムの枠組み

6.1.1 サーバ環境

図 6.1 に示す本学システムの概要について述べる。SCORM 対応の学習管理システムを開発するにあたっては、OS やプログラミング言語の制約はない。そこで本研究では、SCORM のサンプル環境として利用されている Apache Tomcat サーバ (以下 Tomcat) をベースにシステムを開発した。Tomcat は ApacheProject[17] で開発された Java ベースの Web サーバであり、html などの静的なドキュメントの公開はもちろん、動的に Web ページを生成するようなプログラムも jsp や Servlet という形式で作成することができる。そこで、Tomcat をベースとし学習者情報を管理するためのモジュール群を作成し、実際に学習者情報を格納するためのリレーショナルデータベースをモジュールより利用する。教材情報管理モジュール群は、ディレクトリに置かれる教材情報の情報を扱う。さらに SCORM で定義される XML データを XML データベースを用いて管理する。これら、学習者情報管理モジュールと、教材情報管理モジュールの両方を統括するモジュールとして、情報管理モジュールを作成する。その情報管理モジュールを利用することにより、ナビゲーション機能から効率よくデータの取得が行えるような仕組みとなる。

ナビゲーション機能は、図 6.1 に示されるように、学習状態モジュール群とナビゲーション戦略モジュールによって構成される。各学習状態モジュールが必要となる学習者、教材情報を取得するために情報管理モジュールへとアクセスし候補となる SCO を選択する。選択した SCO をナビゲーション戦略モジュールへと渡し、ナビゲーション戦略モジュールが学習者にとって最適である index を作成し提供する形となる。

6.1.2 クライアント環境

本システムではクライアントとして、まず教材閲覧に利用する Web ブラウザが必要となる。ただし Web ブラウザのみでは教材の再生が不可能であるので、各種プラグインが必要となる。まず、本学システムで提供される動画ファイルである RealMovie ファイルを再生させるために必要なプレイヤーである、RealPlayer が必要となる [18]。次に、スライドとして表示される FLASH ファイル再生用の FLASH プレイヤーが必須になる [19]。これらは Web 上から無償で取得できるプラグインである。

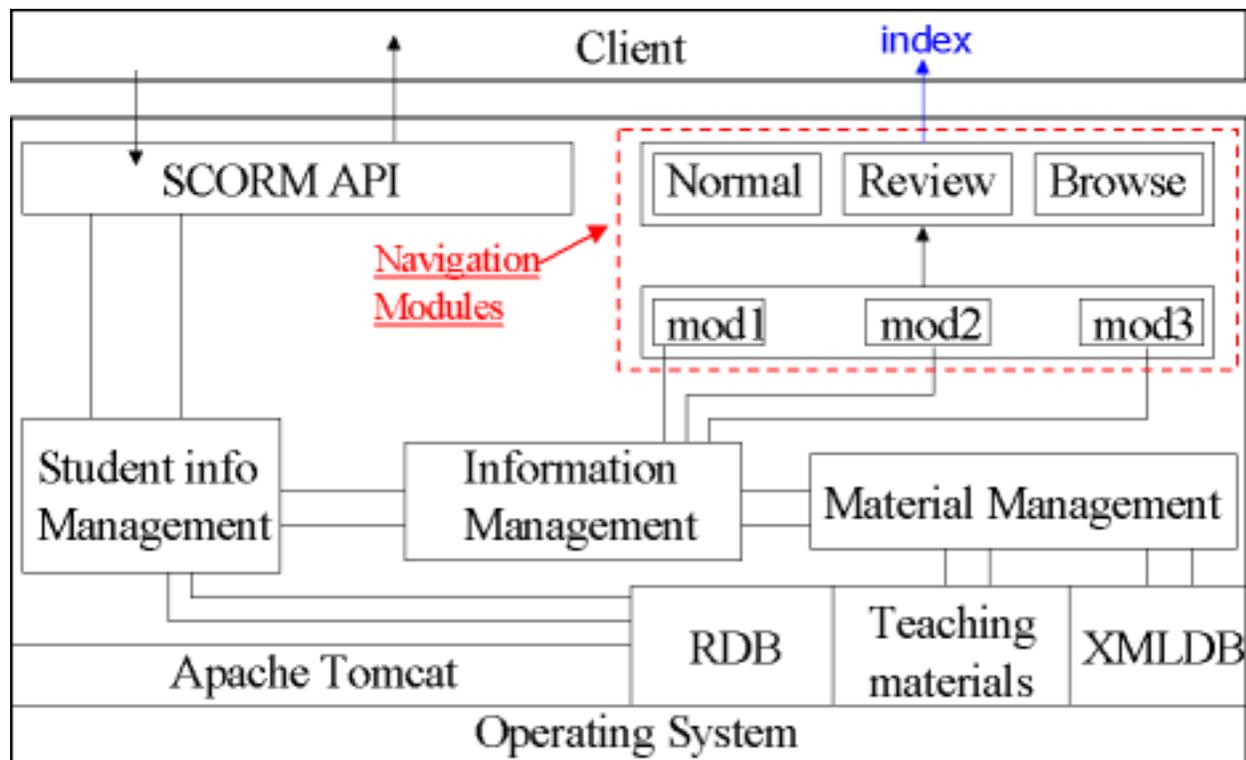


図 6.1: システムイメージ

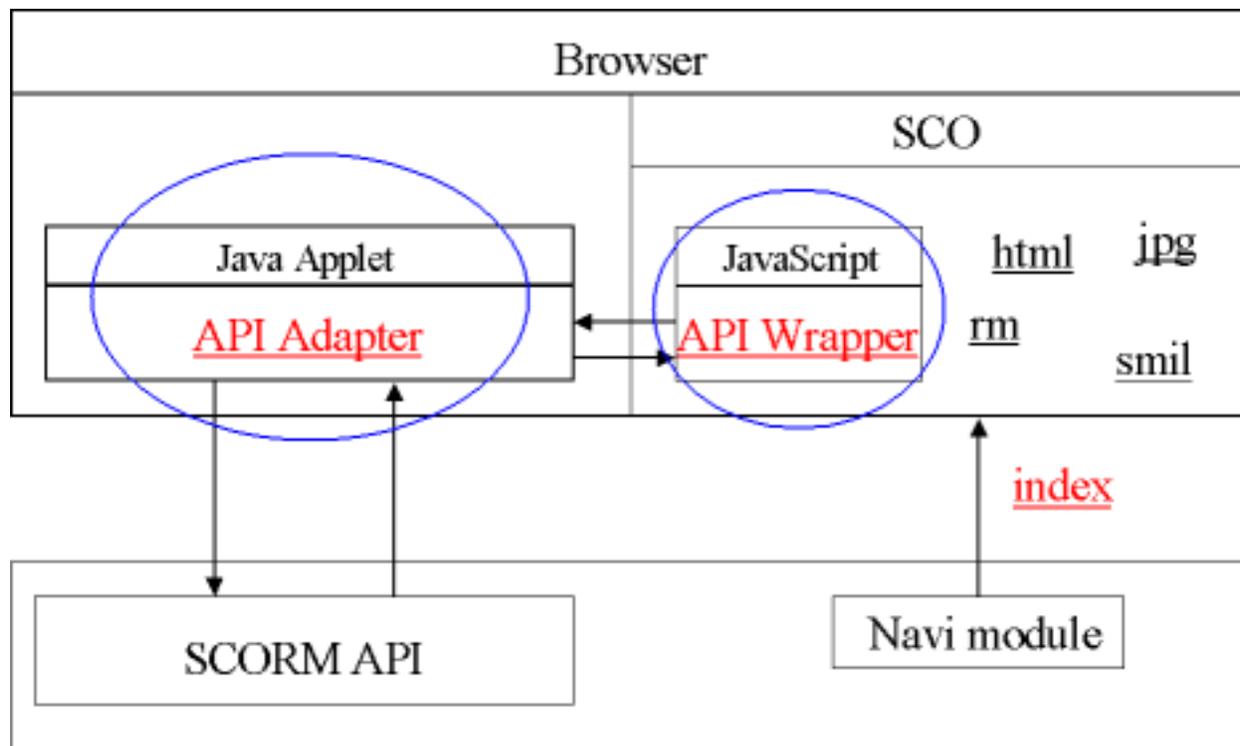


図 6.2: クライアントイメージ

また、図 6.2 で示すように、JavaApplet からなる API Adapter を通じて学習管理システムとの通信を行うので、JavaApplet が実行可能なクライアントサイドの Java Virtual Machine、または Java Runtime Environment が必要となる。さらに、JavaApplet は SCO に含まれる JavaScript を経由して学習管理システムと通信を行うために、JavaScript が動作するブラウザが必須となる。ここで与えられる API Adapter であるが、学習管理システムにログインした時に、学習者環境へと読み込まれ、学習者が学習管理システムを利用した学習を終了し、ログアウトする時点まで学習者に利用される。JavaScript は提供される各々の学習項目である SCO に組み込まれる形となり、学習者が提供された SCO をどれだけ学習しているのかなどの時間をカウントしたり、学習に関する情報を API Adapter 経由で学習管理システムへと渡し、必要な処理を行い学習者履歴情報として処理するような形となる。

6.2 XML ファイル作成モジュール

SCORM で利用されるメタデータは膨大であり、手作業でメタデータファイル作成を行うことは現実的ではない。そこで本研究において、設定ファイルを読み込むことで、ファイル作成作業の付加を軽減できるツールとし、定義されるタグが書き込まれた XML ファイルを出力するモジュールを作成した。

以下に示すような設定ファイルを読み込むことで、XML 文章を生成することができる。

```
general:title:en:chap0101
general:language:Japanese
general:description:ja:SCO を総合的に表示させるページ
general:keyword:jp:自然言語
general:structure:Collection
general:aggregationlevel:2

lifecycle:status:Final
lifecycle:contribute:role:Author:centity:JAIST:date:2002/11/01:Date
.....
```

この設定ファイルは以下の定義となっている。

要素のタグ名:子要素のタグ名:それぞれのタグのオプション:値:...

ここで先頭の「要素のタグ名」は、SCORM のメタデータで定義される 9 つのカテゴリの値である。次にくる「子要素のタグ名」は各カテゴリごとに定義される要素である。それぞれの要素により、取りうるオプションとタグの要素が変わるためすべて同じ値ではない。実際に生成される XML ファイルを図 6.3 に示す。

この XML ファイル生成モジュールを利用することで、SCORM に必要な教材情報であるメタデータを定義する XML ファイルを作成作業を容易にすることができる。

```

<?xml version="1.0" encoding="Shift_JIS" ?>
- <lom>
- <general>
- <title>
  <langstring xml:lang="en">chap0101</langstring>
</title>
  <language>Japanese</language>
- <description>
  <langstring xml:lang="ja">SCOを総合的に表示させるページ</langstring>
</description>
- <keyword>
  <langstring xml:lang="jp">自然言語</langstring>
</keyword>
- <structure>
- <vocabulary>
- <source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
</source>
- <value>
  <langstring xml:lang="x-none">Collection</langstring>
</value>
</vocabulary>
</structure>
+ <aggregationlevel>
</general>
+ <lifecycle>
+ <metametadata>
+ <technical>
+ <educational>
+ <rights>
+ <relation>
  <annotation />
  <classification />
</lom>

```

図 6.3: 生成された XML ファイル

6.3 XML データベース

SCORM で利用されるメタデータの要素は付録で示すように非常に細かく定義されているが、利用できるタグの数は各要素によって可変的である。例えば $\langle title \rangle$ タグに対しては必ず一意のデータが必要であるが、 $\langle keyword \rangle$ タグであれば 0 ~ 20 個まで書き込むことが出来る。またタグが再帰的な構造になる要素も存在する。このように構造が変化する場合は、従来からよく利用されてきたリレーショナルデータベースでは扱いにくい。そこで本研究では、XML ファイルの構造を気にせずに利用できる手法として、ネイティブ XML データベースを利用する。データを XML フォーマットで管理することにより、規格のバージョンアップなどで XML ファイルの構造が変わったとしても、リレーショナルデータベースのようにテーブル属性の変更は必要ない。

本学システムでは、ネイティブ XML データベースとして、The Apache Software Foundation のプロジェクトの 1 つとして開発されている Xindice[20] を利用した。特徴として以下のものが挙げられる。

- (1) Well-Formed ドキュメントを格納可能
- (2) 問い合わせ言語は XPath
- (3) 更新言語は XML:DB XUpdate
- (4) エlement や属性に index を付けることが可能
- (5) Java 用の API が用意されている
- (6) すべて Java で記述されている
- (7) セキュリティ関連の機能はない

(1) は何かといえば、スキーマを定義することなく XML 文章を格納することができるということである。通常、リレーショナルデータベースに XML 文章を格納する場合は、XML 文章を細かく分解して、分解したデータをカラムにマッピングする必要があるため、格納する XML 文章のスキーマをあらかじめ定義しておくことが不可欠であった。しかし、ネイティブ XML データベースはその必要がない。

(2),(3) は、XML データベースで標準という訳ではないが多く利用されている規格である。

(4) は問い合わせを高速化する規格である。

また (5),(6) で示されるように、本研究にて開発される学習管理システムは Java ベースである。つまり、Java 用の API が定義されていることにより、サーバサイドで動作する Java プログラムから比較的簡単にアクセスすることが可能となり、本システムとの親和性が高い。

(7) はユーザ認証といったセキュリティの機能が付いていないので利用する時に注意が必要になるということである [21] .

Xindice では、XML 文章を図 6.4 に示すファイルシステムに似たモデルで管理する。一般にファイルシステムはディレクトリとファイルから構成されるが、Xindice では、ディレクトリに対応する物を「コレクション」、ファイルに対応する物を「ドキュメント」と呼ぶ。本システムでは、階層的なディレクトリ構造に管理される教材群と同様に、SCORM にて定義される Asset メタデータや SCO メタデータを、Xindice を利用して階層構造で管理している。

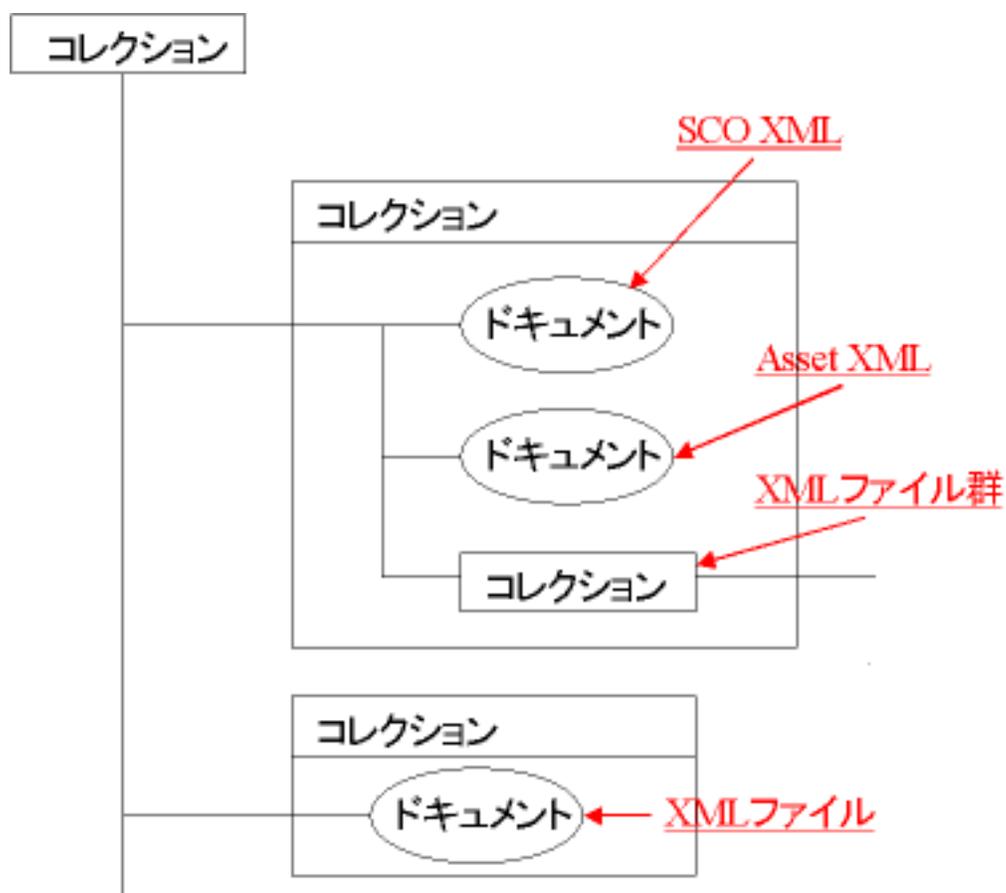


図 6.4: Xindice における文章管理

第7章 ナビゲーションの実行

7.1 学習を進める順序

本システムを用いて学習を進める場合は、以下の手順従うこととなる。

1. システムへのログイン
2. 学習するコースの選択
3. 学習モードの選択
4. 提供される index からの学習項目の選択
5. 選択した学習項目の学習の開始
6. システムのログアウト

次ページより、各ステップを画像と照らし合わせて解説する。

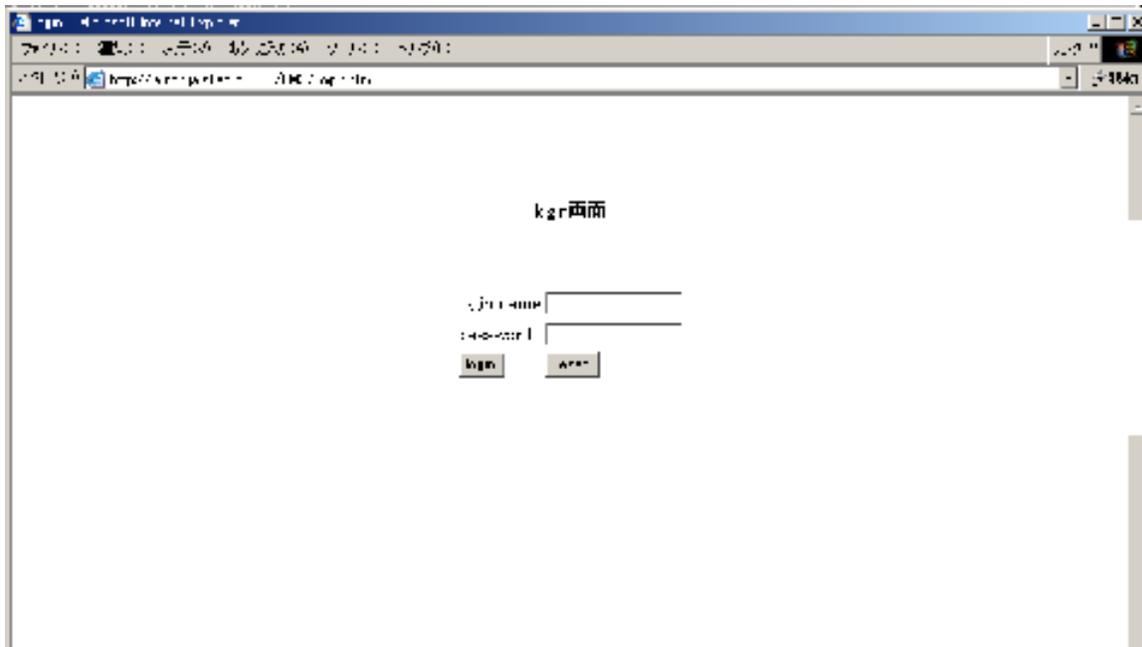


図 7.1: ログイン画面

1. のシステムのログインであるが、図 7.1 で示すように、ユーザ ID とパスワードによる認証を行うことにより、学習管理システムへのログインを行うことができる。本研究では実現しなかったが、NIS(Network Information Service) の情報を利用することにより、学内、または社内などの組織で集中管理されるネットワークとの連携が可能となる。

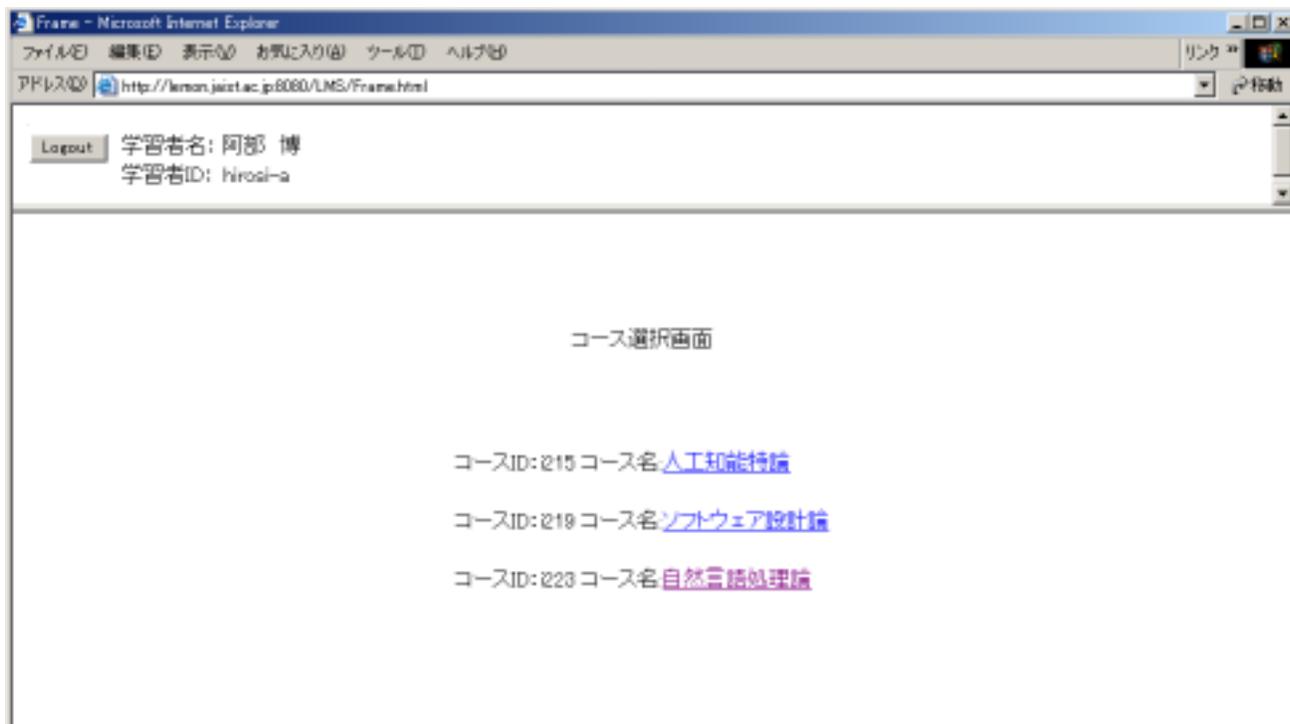


図 7.2: コース選択画面

2. において、ログインした学習者にまず提供されるページは、図 7.2 で示されるコース選択の画面である。ここに表示される情報は学習者が履修しているコースの情報であり、学習者が学習したい科目を選択し学習を進めることができる。将来的に本学で提供される全科目が電子教材として提供されることになれば、この部分において大学が管理する学生の履修情報と連携することが可能となる。

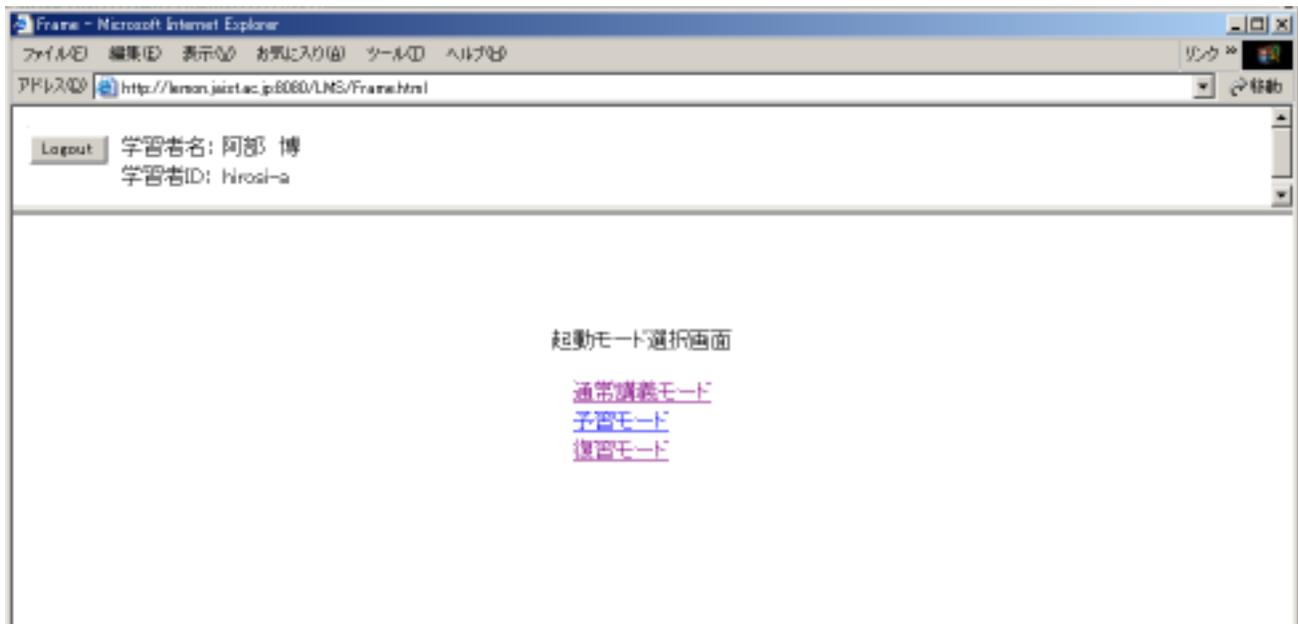


図 7.3: 学習モード選択画面

3. で学習者に提供される画面は、図 7.3 のような学習者がどのような目的で学習を進めるかを決定する学習モード選択画面である。ここで学習者は大まかな学習目標を決め、学習を進めることとなる。



図 7.4: 通常講義モード

4. では、学習者に対して、3. で選択された大まかな学習目標に基づく index が作成され、学習者へと提供される。図 7.4 では、学習者が通常講義モードを選択した場合であり、次に学習すべき項目が強調表示されている。この例では、章節構造に基づき線形に学習を進めた場合であり、学習者に対して「構文解析アルゴリズム 1」が強調されている。このように学習者は、提示される index の中から、学習したいものを選択し、学習を行うことになる。



図 7.5: 教材学習時イメージ

5. では, 図 7.5 で示されるように学習者が選択した学習項目の学習が開始される. 学習者は学習をいつでも中断することができる. その場合は教材に表示される「Back」ボタンを押すことにより, 4. で表示される index 画面へと戻ることができる.

6. は、学習者が学習を終えたい時に「logout」ボタンを押すことでシステムをログアウトすることができることを示す。ログアウト後は、1. で示したログイン画面が表示されることとなる。

このように、1 から 6 の手順を踏むことで、本システムを利用した学習を進めることが可能となる。

第8章 本システムの改善に向けて

本章では、本システムの改善点を指摘し、実現への指針を示す。

8.1 ナビゲーションの単位

現在、学習者に提供されるナビゲーションの単位は、本学の教材構造でいうところの節にあたる部分である。しかしながら実際に学習者が求めるであろうナビゲーションの粒度は、スライドの何枚目の部分の説明であるとか、動画の3分経過後の説明などといったさらに小さな単位であることも多いと思われる。本システムでは、このようにスライドに関連する部分の時間を教材情報より抜き出し、それに関連する動画を再生するような構造は持ち合わせていない。しかしながら実際にナビゲーションを行う場合には、このようなピンポイントに学習者が求める部分を学習者に提示することも効果的であると考えられる。そこで、現在行っているSCOの単位でのナビゲーションのように大きな学習項目単位ではなく、スライド1枚1枚の情報であるAssetレベルの教材情報に動画の開始時間の情報を持たせることで、該当する動画の頭出しを行い学習者が求める部分をピンポイントに学習できるようなシステムを実現すべきである。

8.2 ユーザインタフェース

ナビゲーションを行う際にユーザインタフェースは重要な要素であると思われる。本研究で学習者に提供したユーザインタフェースは必ずしも最適であるとはいえない。学習者が取得できる情報はユーザインタフェースからのみでありそれを用いて学習を進めるのであれば、学習者にとって利用しやすい最適なユーザインタフェースの設計を検討する必要がある。

8.3 学習者履歴の活用

本論文で利用した学習履歴は、学習時間と試験の得点のみであった。それらの情報以外に、学習者自身がどのように学習項目を学習してきたかという学習の順序の履歴など他にも様々な学習履歴情報が活用できる。さらに、他の学習者の履歴情報の利用を検討することも可能である。学習者が初めて学習を行う場合には、システムに提示される学習項目を

学ぶわけであるが、ある学習者がある学習項目でつまずき、そこをどのように打開したかという履歴は他の学習者が乗り越えた方法をつまずく前に効率よく取得することになり、学習者の学習効率を向上するさせる点で有用であると考えられる。

その他にも、全学習者の学習時間が相対的に多い場合は、教材の難易度が高いであるとか、テストの平均点が著しく悪いのであれば、教材の内容が難しい、あるいはテストの難易度が高かったと評価することができ、次の教材作成、改良への情報にもなり得る。

また、学習者の属性を記述するための規格としてLIP(Learning Information Package)[22]がある。主要な項目として、氏名やIDなどの識別情報、学習目的情報、保有資格情報、学習履歴情報、スキル、知識などのコンピテンシ情報、パスワードなどのセキュリティ情報などが含まれている。LIP規格によりシステム間で学習者情報を交換するためのフォーマットが標準化される。教材情報だけでなく、学習者情報にも標準規格を用いることで、システムの汎用化は進むと思われる。

8.4 教材の分散管理

本研究で、学習者に提供される教材は本学で作成された教材にSCORMの情報に基づき、教材構造を与えメタデータを付加したものである。しかしながら本研究で作成されたシステムおよび教材は、本学でのみの配信しか考えられていない。これではせっかくSCORMに基づき教材、ならびにシステムを作成した利点が活かされていないことになる。SCORMに基づくシステムを作成したのであれば、他のSCORM教材も再生できるのである。本学のみならず、電子教材を作成している組織は多数ある。理想的には、他の組織の電子教材をCD-ROMやDVD-ROMなどを利用してインポートするのではなく、インターネットを利用して取得できるような機構が望ましい。

このようなサービスを行うには、どこにどのような電子教材があるのかということを知る必要があり、さらにそれらの情報を共有することにより、情報を必要とするクライアントに利用されやすい環境となる。このような技術はここ数年で発展してきたWebサービス[23]と呼ばれる技術であり、SOAP[24]と呼ばれるプロトコルを用いてHTTP上に実装されている。Webサービスについて詳しく説明すると本研究のテーマよりそれることとなり割愛するが、この技術を用いることにより、分散された電子教材をインターネットを利用して学習することができる環境が提供できる。この技術を確立するためには、著作権の問題や組織間の連携など多数の大きな問題が存在するが、学習者に取って非常に魅力的な学習環境になるとと思われる。

8.5 ナビゲーション機能のライブラリ化

本研究でナビゲーション機能を実現する上で、汎用性と拡張性ということを考えてきた。特にこの2つを考える上で、今以上にさらに共通の部品を細かく分けることで、ソフ

トウェアの部品化が促進する．それらはナビゲーション機能を構築する上でも1つ1つの部品となり，最終的にはライブラリ化することで，システム開発者にとって有益なライブラリ集となるであろう．

第9章 おわりに

9.1 まとめ

本稿では，体系化された電子教材をコースウェアとして持つ遠隔学習システムにおいて，学習者の多様なニーズに合わせた学習項目を提示するナビゲーション機能の枠組みを提案した．また，ナビゲーション機能を効果的に実現するための手段として，本学の遠隔学習システムを例に，SCORM に基づく学習管理システムにおけるナビゲーション機能について検討し実現した．

現在実現しているシステムはナビゲーション機能を持つ学習管理システムのみであるが，我々の研究室ではこの他に電子教材の評価 [25] やオーサリングツールに関する研究を行っている．これらを統合することにより，学習者だけでなく教授者に対するより有益な情報提示を可能にするシステムを実現することを今後の目標とする．さらに，学内ですでに稼働している成績管理システム，カリキュラム管理システムなどと連動し，実際の業務との提携を視野に入れたシステムの実現を今後の課題とする．

9.2 今後の課題

9.2.1 本システムの改善

第8章で示したシステム改善点を実現し，その有効性を評価する必要がある．学習する側である学習者，教材作成側である教授者ならびに教材作成者，さらにはシステム管理，開発者のそれぞれの意見を取り入れ反映させることで，システムならびにナビゲーション機能を発展させる必要がある．

9.2.2 実証実験

本システムは実際に学習者に提供される形でまだ実証実験を行っていない．実際に利用してもらうことにより，システムの的な弱点も浮彫りになると思われる．実証実験による評価を行うことで，有益な知見が得られると考えられる．

謝辞

最後に、本研究を行なうにあたり、終始御指導頂きました落水浩一郎教授に心より感謝申し上げます。

日頃から研究に協力していただき大変有益な御助言をいただきました情報科学センターの長谷川忍助手に深く感謝いたします。

研究ならびに学生生活に関して相談に乗っていただき、助言をいただきました元落水研究室助手故村越広享博士に深く感謝いたし、心よりご冥福をお祈りいたします。

また、論文審査にあたりまして適切な御助言、御意見を賜りました丹康雄助教授、並びに権藤克彦助教授に深く感謝申し上げます。

そして、本研究に関して多くの有意義な助言を頂きました落水研究室の皆様方に感謝申し上げます。

さらに大学時代に情報科学への道を示して頂き数々の助言をも頂きました岩村昌司氏、ならびに岩村美和氏に心より感謝いたします。

最後に、進学への援助をしてくださった両親家族、ならびに私生活の面でお世話になった友人に深く感謝申し上げます。

参考文献

- [1] 伊藤 健二, ”e-Learning とは何か”, 情報処理学会, 情報処理 Vol43 No.4, pp.394-400, Apr. 2002
- [2] 仲林 清, ”e-Learning の要素技術と標準化”, 情報処理学会, 情報処理 Vol43 No.4, pp.401-406, Apr. 2002
- [3] 矢野 米雄, ”教育工学辞典”, pp.73-74, 日本教育工学会編, 実教出版株式会社, 2000.
- [4] 岡本 敏雄, ”教育工学辞典”, pp.6-7, 日本教育工学会編, 実教出版株式会社, 2000.
- [5] 樋川 和伸, ”教育システム情報ハンドブック”, pp.181-184, 教育システム情報学会編, 実教出版株式会社, 2001.
- [6] AICC, <http://www.aicc.org/>
- [7] IEEE Learning Technology Standards Committee(LTSC), IEEE P1484.12 Learning Object Metadata Working Group, <http://ltsc.ieee.org/wg12/index.html>
- [8] IEEE Learning Technology Standards Committee(LTSC), <http://ltsc.ieee.org/>
- [9] IMS, <http://www.imsglobal.org/>
- [10] ADLNet, The SCORM Overview, 2001.
- [11] ADLNet, <http://adlnet.org/>
- [12] e-Learning Consortium Japan, ”SCORM 対応プラットフォーム , オーサリング製品”, <http://www.elc.or.jp/>
- [13] ADLNet, The SCORM Content-Aggregation Model, 2001.
- [14] ARIADNE Educational Metadata Recommendation, http://ariadne.unil.ch/Metadata/ariadne_metadata_v3final1.htm
- [15] ADLNet, The SCORM Runtime-Environment, 2001.

- [16] 結城 浩, ”Java 言語で学ぶデザインパターン入門”, ソフトバンクパブリッシング株式会社, 2001 .
- [17] The Apache Jakarta Project, <http://jakarta.apache.org/tomcat/index.html>
- [18] Real Networks, <http://www.real.com/>
- [19] Macromedia, <http://www.macromedia.com/>
- [20] Apache Xindice, <http://xml.apache.org/xindice/>
- [21] 野 畠 英 明, ”Xindice : 無 料 で 使 え る XML デ ー タ ベ ー ス”,
<http://www.atmarkit.co.jp/fxml/tanpatsu/18xindice/xindice01.html>
- [22] IMS Learner Information Packaging Information Model Specification,
<http://www.imsglobal.org/profiles/lipinfo01.html>
- [23] Web Service Activity, <http://www.w3.org/2002/ws/>
- [24] SOAP Version 1.2 Part 1: Messaging Framework, <http://www.w3.org/TR/soap12-part1/>
- [25] 猪俣敦夫, ”遠隔学習における電子教材の AHP を利用した評価法に関する研究”, 北陸先端科学技術大学院大学 情報科学研究科 博士論文 2002

本研究に関する発表論文

- [1] 阿部 博，長谷川 忍，落水 浩一郎，”ナビゲーション機能を有する学習管理システムに関する研究”，電子情報通信学会，ET2002-84 94,pp.7-12,January 2003.

付録

SCORM にて定義されるメタデータ集

SCORM で定義されるメタデータをカテゴリごとに表で示す。

表 1: General カテゴリ

| 各要素 | 各要素に関する情報 |
|------------------|-----------------------------|
| general | 学習者の特性や特徴を統括する一般情報 |
| title | 学習資源に与えられた名前 |
| catalogentry | 学習資源に割り当てられたカタログへのエントリ |
| catalog | カタログの名前 |
| entry | カタログ中のエントリの実際の文字列値 |
| language | 学習資源をやりとりする時に使用する言語 |
| description | 学習資源の内容についての文書記述 |
| keyword | 特徴を表すキーワードやフレーズ |
| coverage | 学習資源が適用される時間，文化，地理，分野の範囲や期間 |
| structure | 学習資源の基本的な構造 (集中型，階層型，直線型など) |
| aggregationlevel | 学習資源の機能的な粒度 |

表 2: Lifecycle カテゴリ

| | |
|------------|---------------------------------|
| lifecycle | このカテゴリは，資源の経歴や現状，製作に貢献した人物を記述する |
| version | 資源のバージョン |
| status | 資源の状態，状況 |
| contribute | 資源の開発に貢献した人物や組織 |
| role | 貢献の種類 |
| entity | 資源に貢献した人や組織についての情報の識別子 |
| date | 貢献した日付を定義 |

表 3: Metametadata カテゴリ

| | |
|----------------|---------------------------------|
| metametadata | メタデータレコード自身に関する詳細情報 |
| catalogentry | メタデータインスタンスに与えられたカタログ |
| catalog | カタログの名前 |
| entry | カタログ中の実際の値 |
| contributor | メタデータインスタンス作成に貢献した人や組織 |
| role | 貢献の種類 |
| entity | メタデータインスタンスに貢献した人や組織についての情報の識別子 |
| date | 貢献した日付を定義 |
| metadatascheme | メタデータを生成するのに使用した仕様名とバージョン |
| language | メタデータインスタンスの言語 |

表 4: Technical カテゴリ

| | |
|---------------------------|--------------------------|
| technical | 資源の技術的な要求事項や特徴 |
| format | 資源の技術的なデータタイプ |
| size | デジタル資源のサイズ |
| location | 資源の格納場所を表す文字列 |
| type | 格納場所を識別するためのタイプ |
| requirement | 資源を使用する場合の技術的な要件 |
| type | 資源を使用する際に要求される技術 |
| name | 資源を使用する場合に要求される技術の名前 |
| minimumversion | 資源を利用する際に要求される技術の最低バージョン |
| maximumversion | 資源を利用する際に要求される技術の最高バージョン |
| installationremarks | 資源のインストール方法に関する記述 |
| otherplatformrequirements | その他のシステム要件 |
| duration | 資源を想定された速度で再生した場合に要する時間 |

表 5: Educational カテゴリ

| | |
|----------------------|----------------------------|
| educational | 資源のキーとなる教育的または教育学的な特性を記述する |
| interactivitytype | ユーザとのインタラクションのタイプ |
| learningresourcetype | 資源の（主要な）種類 |
| interactivitylevel | ユーザとの対話レベル |
| semanticdensity | 学習効率の主観的なレベル |
| intendedenduserrole | 利用者の種別 |
| context | 利用環境 |
| typicalagerange | 想定されるユーザの年齢層 |
| difficulty | 難易度 |
| typicallearningtime | 標準学習時間 |
| description | 資源をどのように使用するかについてのコメント |
| language | 想定ユーザが使用する言語 |

表 6: Rights カテゴリ

| | |
|----------------------------------|---------------------|
| rights | 資源の著作権や使用条件 |
| cost | 使用する上で使用料が求められるかどうか |
| copyrightandotherresistrications | 著作権および制約 |
| description | 資源の使用条件についてのコメント |

表 7: Relation カテゴリ

| | |
|--------------|--------------|
| relation | 他のオブジェクトとの関連 |
| kind | 対象資源との関係 |
| resource | 対象となる資源 |
| description | 対象資源の記述 |
| catalogentry | カタログエントリ |
| catalog | カタログ名 |
| entry | カタログ中の実際の値 |

表 8: Annotation カテゴリ

| | |
|-------------|------------------|
| annotation | 教育的使用におけるコメントや注釈 |
| person | 注釈の作成者 |
| date | 注釈の作成された日付 |
| description | 注釈の内容 |

表 9: Classification カテゴリ

| | |
|----------------|-----------------------|
| classification | 特定の分類規則でどこに資源が属するかを記述 |
| taxonpath | 分類体系へのパス |
| source | 分類体系の名前 |
| taxon | 各分類項目 |
| id | 各分類項目の識別子 |
| entry | 各分類項目へのテキストラベル |
| description | 特定の分類体系における記述 |
| keyword | 特定の分類におけるキーワード |