

拡張性を有する学習支援システムアーキテクチャに基づく 分散マルチプラットフォーム学習環境の設計と試作

仲林 清^{*1,2} 森本容介^{*3}

*1 千葉工業大学 *2 熊本大学 *3 放送大学

Design and Prototype Implementation of Distributed Multi-Platform Learning Environment based on an Extensible Learning Support System Architecture

Kiyoshi Nakabayashi^{*1,2} Yosuke Morimoto^{*3}

*1 Chiba Institute of Technology *2 Kumamoto University *3 The Open University of Japan

拡張性を有する学習支援システムアーキテクチャ ELECOA に基づく分散マルチプラットフォーム学習環境の検討を進めた。ELECOA の特徴は、プログラムモジュールである教材オブジェクトを構成要素とし、機能拡張性とコンテンツ再利用性を両立させた柔軟な学習環境を提供することにある。これまでに教材オブジェクト間の基本的な通信パターンとこれを遵守・再利用するためのデザインパターンを示し、このデザインパターンで独習型・グループ学習型のさまざまな環境を実現できることを示した。本稿では、このデザインパターンに則って、教材オブジェクトを Web サーバやスマートフォンなど複数プラットフォームに動的に再配置する方式、および、実用的な学習支援環境への適用について述べる。

キーワード: 学習者適応, 拡張可能学習支援システム, 教材オブジェクト, 分散プラットフォーム

1. はじめに

e ラーニング教材や教授方略を共有・流通・再利用するため、さまざまな標準規格が開発されている⁽¹⁾。教材については、SCORM 規格⁽²⁾、QTI 規格⁽³⁾、教授方略については、LD 規格⁽⁴⁾⁽⁵⁾が挙げられる。これらの標準規格に準拠した学習環境では、規格範囲外の機能拡張を行うことは困難である。拡張機能を有するコンテンツは、非対応のプラットフォームでは動作せず、相互運用性が損なわれてしまうためである⁽⁶⁾⁽⁷⁾。

筆者らは、相互運用性と機能拡張性の両立を図るため、教材オブジェクトという概念を導入した拡張可能な学習支援システムアーキテクチャ Extensible Learning Environment with Courseware Object Architecture (ELECOA) を提案してきた⁽⁸⁾⁽⁹⁾。さらに、独習、グループ学習の学習制御に共通する教材オブジェクト間の基本通信パターン⁽¹⁰⁾、教材オブジェクトがこの通信パターンを遵守するためのデザインパターンを導いた⁽¹¹⁾⁽¹²⁾。このデザインパターンに則って教材オブジェクトを実装し、個別に開発された独習用コンテンツをグループ学習で再利用することが可能になること、グループ型作問学習など実用的な学習環境が実現できることを示した⁽¹²⁾。また、近年提案された独習環境規格の cmi5 に適用した⁽¹³⁾。

本稿では、ELECOA のアーキテクチャを踏まえて、再利用可能な教材オブジェクトが、Web サーバやスマートフォンなどに分散・連携するマルチプラットフォーム学習環境⁽¹⁴⁾⁽¹⁵⁾の検討・開発状況を報告する。以下、第 2 章で ELECOA の概要、第 3 章で教材オブジェクト間通信パターン、第 4 章で通信パターンを実装するためのデザインパターンについて述べる、第 5 章で分散マルチプラットフォーム学習環境の概要、第 6 章で教材オブジェクトの動的再配置方式、第 7 章で実用的学習支援環境への適用について述べる。

2. ELECOA の概要

従来の学習支援システムは、コンテンツとプラットフォームを分離する構成が一般的である⁽⁷⁾。学習内容に依存しない共通機能はプラットフォームに実装し、コンテンツ作成者はプラットフォームの実装に関与せずに、学習内容に応じたコンテンツを作成する。このような構成では、プラットフォームに予め決められた機能を実装するため、後からの機能拡張が困難であり、不用意な改造で既存コンテンツが動作しなくなるといった問題が発生する。また、コンテンツとプラットフォームの相互運用のためには標準規格が必要であるが、規格準拠のシステムに独自機能を追加すると、相互運

用性が確保できなくなる。このように、従来の構成では機能拡張性と相互運用性の両立は非常に困難である。

そこで、筆者らは「教材オブジェクト」⁽⁶⁾と呼ぶ概念を取り入れた学習支援システムアーキテクチャ ELECOA⁽⁶⁾⁽⁷⁾を提案した。図 1 に示すように、ELECOA は、コンテンツ、教材オブジェクト、プラットフォームの 3 層の構成になっている。教材オブジェクトは、従来型アーキテクチャでプラットフォームに実装されていた、学習者適応などの機能を取り出したプログラム部品である。機能追加の際は、新規の教材オブジェクトを作成する。既存コンテンツは既存教材オブジェクトのみで動作するため、機能追加の影響を受けない。このため、カスタマイズが格段に容易になり、機能拡張性を向上できる。また、コンテンツと教材オブジェクトを一緒に流通させることで、相互運用性、コンテンツ再利用性を確保できる。

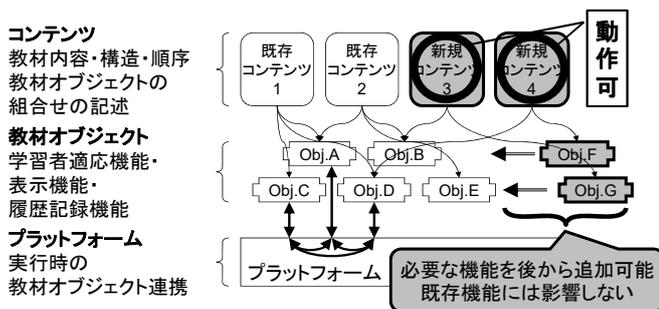


図 1 ELECOA の構成

3. 教材オブジェクト間通信パターン

3.1 通信パターンの機能

ELECOA では階層型コンテンツの各ノードに教材オブジェクトを配置する。木構造の一部のサブツリーを取り出して再利用可能とするため、教材オブジェクト間のもっとも基本的な通信は、直接の親子間に限定する⁽⁷⁾。ツリー全体の学習制御機能は、直接の親子間通信を組み合わせる。このための通信パターンは、独習型コンテンツ⁽⁷⁾で規定された以下の 4 つの処理に対応する⁽¹¹⁾⁽¹²⁾。

- (1) ロールアップ処理
- (2) ポストコンディショナルルール処理
- (3) シーケンシング処理
- (4) 学習コマンドリスト生成処理

グループ学習環境でも、各学習者に階層型コンテンツを割り当て、教材オブジェクトが、他学習者の教材

オブジェクトと情報交換することで学習制御を行う。グループ学習では、「他学習者の状態を条件とする分岐」、「他学習者との同期（待ち）」、「他学習者の状態を条件とする強制移動」が必要となる。これらのうち、分岐は上記(3)のシーケンシング処理で、同期と強制移動は(2)のポストコンディショナルルール処理で実現できる⁽¹¹⁾⁽¹²⁾。このように、上記の 4 つの通信パターンで、独習型コンテンツ・グループ学習環境のいずれも実現することができ、これらの通信パターンを遵守している教材オブジェクトは自由な組み合わせが可能となる。

3.2 通信パターンの実装

前節で述べた 4 つの処理に対応した通信パターンは、いずれも、ツリーの直接の親子の教材オブジェクト間の通信で実現される。また、いずれも、学習者に提示されている学習資源に対応する葉の教材オブジェクト（これをカレントオブジェクトと呼ぶ）が通信の起点となり、親の教材オブジェクトに向けて順次コマンドを送り、それに対する親の応答が子が受け取るという動作を行う。それぞれの教材オブジェクトでは、コマンドに対する処理を独自に実装することができ、さらに新たなコマンドを追加することができる。これによって ELECOA の機能拡張性が実現される⁽⁷⁾。

4 つの通信パターンは、前節に示した(1)~(4)の順で実行される。ツリーのどのレベルのノードまで通信が伝搬するかは、処理によって異なる。ポストコンディショナルルール処理では、下位のコンテンツを再利用しつつ大域的な学習制御を可能とするため、必ずルートまで伝搬が行われ、最もルートに近いノードのルールが優先される。一方、シーケンシング処理では、配下のサブツリー内に次の提示候補となる葉ノードが見つかった教材オブジェクトで伝搬は終了し、カレントオブジェクトに提示候補のノード指定情報が返却される。

4. 教材オブジェクトのデザインパターン

ELECOA の機能拡張性やコンテンツ再利用性を実現するためには、すべての教材オブジェクトが前章の通信パターンを遵守する必要がある。この通信パターンは、以下の処理レイヤーに分離することができる。

- (1) 隣接する親子教材オブジェクト間の通信処理
- (2) (1)の隣接通信を組み合わせた 4 つの通信パター

ンの処理

(3) (2)の 4 つの通信パターンのコマンドに対する個々の教材オブジェクトの処理

そこで、図 2 に示すように、各処理レベルに対応した層を設けたレイヤーデザインパターン⁽¹⁶⁾⁽¹⁷⁾を適用する⁽¹¹⁾⁽¹²⁾。このデザインパターンは、「隣接通信層」、「基本通信層」、「アプリケーション層」から構成され、それぞれが、上記の(1), (2), (3)の処理に対応している。左端の Command Entry は、学習者がコマンドを入力すると、カレントオブジェクトに対して、順次 4 つの通信パターンを起動するコマンドを送り、これが葉ノードからルートまで伝搬して学習制御処理が実行される。このように、レイヤーデザインパターンを適用し、「隣接通信層」、「基本通信層」は各教材オブジェクトで共通とし、「アプリケーション層」で独自機能を実装することで、機能の異なるすべての教材オブジェクトが通信パターンを遵守することができる⁽¹¹⁾⁽¹²⁾。

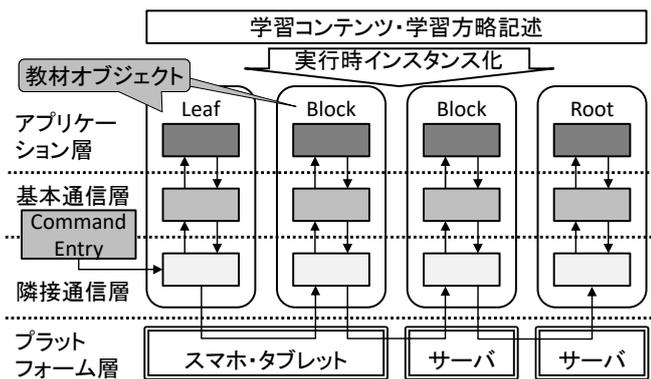


図 2 レイヤーアーキテクチャにおける物理的な通信の隠ぺい

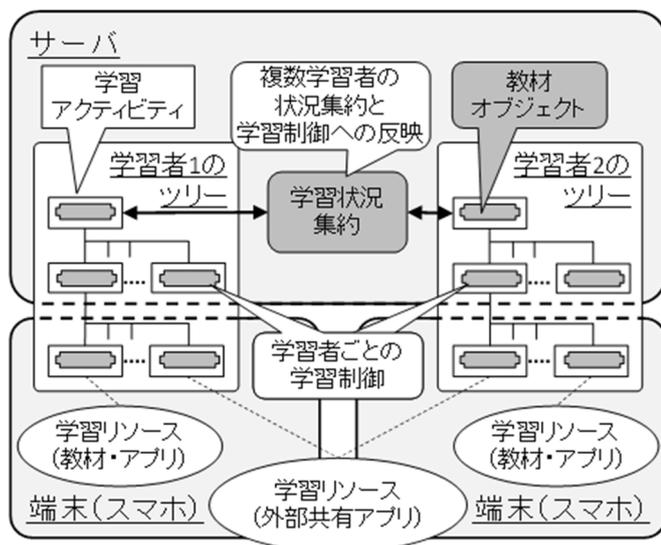


図 3 分散マルチプラットフォーム学習環境

5. 分散マルチプラットフォーム学習環境

5.1 概要

ここまで、4 つの通信パターンを遵守した教材オブジェクトの組合せで、独習やグループ学習などさまざまな学習制御が可能となることを示した。さらに、これらの学習制御を実装するための教材オブジェクトのデザインパターンを示した。このデザインパターンは、図 2 のようなレイヤーデザインパターンであり、隣接教材オブジェクト間の通信を司る隣接通信層と各教材オブジェクトの学習制御を実装する基本通信層、アプリケーション層が明確に分かれている。従って、図 2 のように物理的な通信手段を上位層から隠ぺいすれば、教材オブジェクトとして実装されたさまざまな学習制御機能がサーバ、スマートフォン、電子教科書など、複数プラットフォームで再利用可能な分散マルチプラットフォーム学習環境を実現できる⁽¹⁴⁾⁽¹⁵⁾。

この学習環境の構成イメージを図 3 に示す。コンテンツツリーは、サーバとスマートフォンなどの端末にまたがって配置される。通常は、端末がサーバと通信してグループ学習のための学習制御が行われる。端末がオフラインになった場合も、端末上の教材オブジェクトによって独習を進めることができる。

このような分散マルチプラットフォーム学習環境を実現するうえで検討すべき課題を挙げる。

- (1) 分散プラットフォームアーキテクチャ
- (2) 教材オブジェクト間通信削減方式
- (3) 教材オブジェクト動的再配置方式
- (4) 実用的学習支援環境への適用

本章では、(1), (2)を簡単に紹介し、以降の章で、(3), (4)について述べる。

5.2 分散プラットフォームアーキテクチャ

前節で述べたように、教材オブジェクトの複数プラットフォームにまたがる流通・再利用性を確保するために、上位層から下位層の物理実装を隠ぺいする。図 2 の隣接通信層とプラットフォーム層の間に、隣接通信のための API を設ける。教材オブジェクトが通信を行う際には、通信相手の教材オブジェクトの論理 ID を指定して、この API を起動する。プラットフォーム層では、教材オブジェクトの論理 ID と教材オブジェク

トが配置されたプラットフォームのマッピングを保持している。これによって、プラットフォーム層は、通信相手の教材オブジェクトが、自プラットフォーム内のものか、他プラットフォーム上のものかを判断し、通信メッセージを転送する。これによって、教材オブジェクトの隣接通信層以上は、通信がプラットフォームをまたがっているか、プラットフォーム内に閉じているかを関知する必要がなくなる。

プラットフォーム層の制御プログラムは、教材起動時に、サーバ、スマートフォンなどの物理的なプラットフォームごとに配置され、WebSocket による相互通信の確立、および、教材オブジェクトのマッピングを行う。以後の教材オブジェクト間通信は、このWebSocket を通じて行われる。

教材オブジェクトの実装言語には、Web サーバ、ブラウザ、スマートフォンなどで共通に実行可能なJavaScript を選択した。これによって、同一の教材オブジェクトがいずれのプラットフォームでも動作可能となる。

5.3 教材オブジェクト間通信削減方式

教材オブジェクトは複数プラットフォームに分散配置されるため、その間の通信オーバーヘッドが問題となる。例えば、ポストコンディションルール処理では、最もルートに近いノードのルールを優先するために、必ずルートまで伝搬が行われる。このため、コンテンツ内の移動がひとつのプラットフォームに閉じていても、プラットフォームをまたぐ通信が発生する。そこで、教材オブジェクト間の通信パターンが定型化されていることを活用して通信オーバーヘッドを抑制できる通信方式の検討と削減効果の評価を行っている⁽¹⁵⁾⁽¹⁸⁾。

6. 教材オブジェクト動的再配置方式

携帯端末や電子教科書は、学校内外で通信環境が異なることが考えられる。一方、5.2 節で述べたように、プラットフォーム層は、隣接通信層以上から隠ぺいされており、教材オブジェクトはどのプラットフォームでも動作する。これを活かして、教材オブジェクトを動作させるプラットフォームを動的に変更する方式を検討する。具体的には、教室内のように学校サーバと直接通信が可能な環境では、グループ学習などが円滑

に実施できるように、サーバ上で教材オブジェクトを動作させて学習者間の通信オーバーヘッドを低減する。家庭や移動中など、サーバと通信できない環境で学習を継続するためには、サーバ上の独習教材部分を携帯端末にダウンロードし、後で学習状態をサーバと同期する方式を検討する。

教材オブジェクトが複数プラットフォームに分散配置されている場合、プラットフォーム間での通信が可能な場合でも通信を抑制することが望ましく、オフライン環境の場合は、通信を遮断しても学習が継続できる必要がある。一般的な通信量の削減については、キャッシュなどを用いる方式を提案した⁽¹⁵⁾⁽¹⁸⁾。ここでは、図3に示したような分散マルチプラットフォーム学習環境で、スマートフォンなどに配置されたコンテンツツリーの一部が、他のプラットフォームに配置されたコンテンツ部分と通信を行わずに学習動作を継続できる条件とそのための方策を検討する。また、それに基づくオフラインでの学習継続のための教材オブジェクト動的再配置方式について述べる。

6.1 プラットフォーム間通信の分類

プラットフォーム間の通信は、コンテンツツリー上の親子間のパス、および、プラットフォームをまたがる共有学習目標を通じた通信で生じる。以下、それぞれについて、図4を例に検討していく。

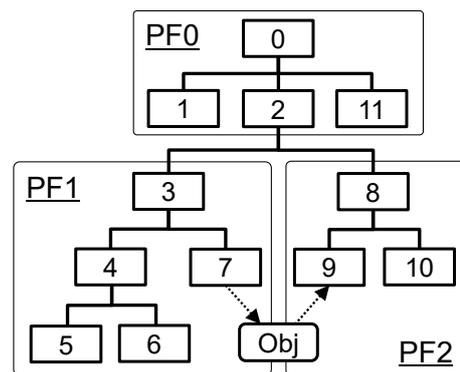


図4 プラットフォームをまたがる共有学習目標の例

6.1.1 コンテンツツリー上の親子間パス

3.1 の4つの通信パターン⁽¹¹⁾⁽¹²⁾のうち、「(1) ロールアップ処理」、「(2) ポストコンディションルール処理」、「(4) 学習コマンドリスト生成処理」では、末端のカレントオブジェクトからコンテンツツリーの最上位のルートノードを結ぶパス上の親子ノード間で通信が生じ

る。「(3)シーケンシング処理」では、次に提示するノードを探索するため、現在提示されているカレントオブジェクトと次に提示するノードを含むサブツリー内で通信が起きる。以上の各処理について、プラットフォームをまたがる通信が発生する状況を検討する。

(1) ロールアップ処理

学習状態の変化は、カレントオブジェクトからルートノードに向けて順次伝搬する。状態変化が同一プラットフォーム内のノードで留まれば⁽¹⁵⁾⁽¹⁸⁾、プラットフォーム間通信は起きない。例えば、図4でノード5が状態変化してロールアップ処理を行った場合、ノード5→4→3→2→0の順に伝搬が起きるが、4ないし3が状態変化しなければ、その上位への通信は不要でPF1からPF0への通信は起きない。

(2) ポストコンディションルール処理

カレントオブジェクトからルートノードに向けて、ポストコンディションルールが設定されているノードの学習状態を参照して実行される。従って、プラットフォーム外のルートノードまでのパス上のノードにポストコンディションルールが設定されていなければ、プラットフォーム間通信は起きない。例えば、図4でノード2、0にルールが設定されていなければこれらへの通信は不要でPF1からPF0への通信は起きない。また、ルールが設定されていても、そのルールの評価が以前に行われていて条件が成り立たず、その後、当該ノードの状態が変化していなければ、やはり通信は不要である。一方で、ポストコンディションルールは、ルートに近いノードのルールが優先される⁽¹²⁾。従って、図4で、ノード3と0にポストコンディションルールが設定されていて、学習状態が変化した場合、ノード3のルールの成立いかんによらず、ノード0のルールも評価する必要がある。この場合は、PF1からPF0への通信が必要である。

(3) シーケンシング処理

カレントオブジェクトを含むサブツリーの中で、次に提示するカレントオブジェクトを検索する⁽¹²⁾。このとき、サブツリーのルートノードまでの通信が起きる。サブツリー内で条件を満たす提示候補が見つからなければ、一段外側のサブツリーを検索範囲とする。従って、図4の例では、ノード5がカレントオブジェクトとすると、まずノード4以下のサブツリー内を検索し、

提示候補が見つからなければノード3以下のサブツリー内を検索する。ここで提示候補が見つければ、PF1からPF0への通信は発生しない。提示候補が見つからなければ、ノード2以下のサブツリー内を検索する必要があり、プラットフォーム間の通信が発生する。

(4) 学習コマンドリスト生成処理

カレントオブジェクトからルートノードまでの各ノードが、自身に定義された実行可能コマンドをカレントオブジェクトに返却する⁽¹²⁾。各ノードの実行可能コマンドが、ノードの状態に依らず変化しなければ通信は不要である。状態に依存していても状態が変化していなければ通信は不要である。

6.1.2 プラットフォームをまたがる共有学習目標

図4の共有学習目標Objが存在するような場合は、コンテンツツリーに沿った通信以外に、共有学習目標を介した学習状態の伝搬が生じる⁽⁷⁾。このような状態伝搬は、直接的には、ロールアップ処理で発生するが、共有学習目標の変化がさらに他のノードの状態変化を起こして、そのノードに設定されているルールに影響を及ぼすほか、ポストコンディションルールなどは共有学習目標を直接参照することもできるので、それらへの影響も検討が必要である。

(1) ロールアップ処理

学習状態の変化は、共有学習目標を介しても伝搬する。従って、図4のノード7で状態変化が起きた場合、Objを経由して、ノード9→8→2→0のパスのロールアップも発生する。

(2) ポストコンディションルール処理

処理はカレントオブジェクトからルートノードに向けて実行される。そのため、ノード7がカレントオブジェクトの場合、ノード7→3→2→0のパスで処理が行なわれ、Objは無関係である。ただし、(1)のロールアップ処理でノード2、0の状態が変化し、それらのノードにルールが設定されていた場合は、Objを経由した状態変化もルールの評価に影響を及ぼす。

(3) シーケンシング処理

前節で述べたように、処理はカレントオブジェクトを含むサブツリー内で行われる。従ってサブツリーの範囲に、共有学習目標を参照する他のノードが含まなければ影響はない。図4の例では、ノード2以下のサブツリー内を検索するまでは影響はない。

(4) 学習コマンドリスト生成処理

前節(4)で述べたように、処理はカレントオブジェクトからルートノードまでの各ノードが関係する。各ノードの実行可能コマンドが、ノードの状態に依らず変化しなければ通信は不要である。依存していても、ノードが(1)で述べた共有学習目標からのロールアップの範囲外であれば影響を受けない。

6.2 分散プラットフォーム間学習状態伝搬方式

以上の検討に基づき、複数プラットフォーム間で共有学習目標経由の学習状態伝搬を遅延させて、プラットフォーム間の通信を軽減する方式を検討した。

着目した点は、シーケンシング処理における移動先の検索範囲である。探索処理はカレントノードを含むサブツリー内で行われる。このサブツリーがカレントオブジェクトが配置されたプラットフォームに閉じていれば、プラットフォーム間通信は発生しない。

一方、検索範囲が他プラットフォームにまたがる場合は、プラットフォーム間通信は必須である。しかし、このとき、探索範囲のノードの学習状態は、シーケンシング処理の対象となるまでに確定していればよい。従って、図4でObjを経由してPF2およびPF0のロールアップによる状態変化が発生する可能性があっても、シーケンシング処理の検索範囲がPF1に閉じている間は、このロールアップ処理の実行は遅延でき、プラットフォーム間の通信を抑制できる。具体的な処理を図4を例に説明する。

- (1) PF1のロールアップ処理でObjの学習状態が変化した場合これを記録し、PF2への伝搬は行わない。
- (2) シーケンシング処理でPF1からPF2への移動が起きた(PF2が探索範囲に含まれた)場合、シーケンシングコマンドとObjの学習状態をPF2へ送る。
- (3) PF2はObjの学習状態とシーケンシングコマンドを受け取り、まず、Objを起点とするロールアップ処理を実行し、その後、シーケンシングを実行する。すなわち、まずObjにロールアップコマンドを送り、ロールアップ処理が完了してから、ノード8配下のサブツリーでシーケンシング処理を行う。

通常の処理の順序は、「ロールアップ処理」→「ポストコンディショナルルール処理」→「シーケンシング処理」であるので、上記の(3)では、シーケンシング処理

の途中で、ロールアップ処理が入れ子になって実行されていることになる。これによって、PF2でシーケンシング処理が行われるまで、Obj経由のロールアップ処理を遅延できる。なお、(1)~(3)の処理は、教材オブジェクトのデザインパターンに則って各レイヤーの処理が規定されているが、詳細は紙幅の都合で割愛する。

ここで、ポストコンディショナルルール処理は、シーケンシング処理の前のカレントオブジェクト(ノード7)からコンテンツツリーのルートノードへのパス上で実行されるので、PF2中のノードに設定されたポストコンディショナルルールの評価は不要である。しかし、上記のパス上にあるPF0中のノード2,0にポストコンディショナルルールが設定されている場合は、ObjからPF2を経由した学習状態変化がこれらのルール評価に影響を及ぼす。ポストコンディショナルルールが成り立つと、シーケンシング処理の対象コマンド自体が置き換えられるため、ノード2,0にポストコンディショナルルールが設定されている場合は、ロールアップ処理の遅延実行を適用することはできない。ただし、このような共有学習目標を経由した学習状態変化が、上位ノードのポストコンディショナルルールに影響を及ぼすケースは実用的に意味が薄く、実際SCORMのテストケースにもみられない。今回、これに関する議論の詳細は紙幅の都合で割愛する。

同様に、学習コマンドリスト生成処理についても、ノード2,0の実行可能コマンドがノードの学習状態に依存する場合は、ObjからPF2を経由した学習状態変化がこれらのノードの実行可能コマンドリストに影響を及ぼす可能性があるため、ロールアップ処理の遅延実行を適用することはできない。ただし、ほとんどの場合、実行可能コマンドはノードの学習状態に依存しないため、現在の実装では実行可能コマンドリストは初期化時に一度だけ設定している。これについては今後の実装で修正予定である。

6.3 動的再配置方式

前節では、共有学習目標が複数プラットフォームにまたがっている場合に、共有学習目標を経由したロールアップ処理を遅延させて、プラットフォーム間通信を抑制する方式を検討した。本節では、これに加えて、6.1.1で述べたコンテンツツリー上の親子間パスに着

目し、サーバ上のコンテンツの一部をタブレット端末などにダウンロード（再配置）し、端末上でオフラインで学習を継続し、再度、サーバと学習履歴を同期して学習を継続する方式を示す。

図4のコンテンツツリー全体が最初PF0に配置されていて、その後、一部のサブツリーを図のようにPF1に再配置して、オフラインで実行することを考える。ここで、前節で述べたように、PF0中のノード2,0にポストコンディショナルルールが設定されていて、ObjからPF2を経由した学習状態変化がこれらのルールに影響を与える場合は、通信が必要でオフライン実行はできない。そこで、ここでは、ポストコンディショナルルールとObjが同時には存在しないケースを考える。

Objが存在する場合は、前節の議論と同様、PF1外への移動が生じるまでObj経由のロールアップを遅延させることで、プラットフォーム間の通信を行わずにオフラインで学習を継続できる。従って、PF1では、前節と同様の処理を行う。

一方、上位ノード（図4のノード2,0）にポストコンディショナルルールがある場合は、PF1内に閉じた動作中でも、これらのルールがシーケンシングコマンドに影響を与える可能性があるため、ルール評価が不可欠である。また、ポストコンディショナルルールは当該ノードの状態を参照するので、そのノードの状態更新のためにロールアップ処理も必要である。

そこで、6.1.1で述べたように、ロールアップ処理、ポストコンディショナルルール処理が、カレントオブジェクトからルートまでのパス上のみで発生し、かつ、ロールアップ処理による学習状態の更新は、処理を行うノードの子ノードの状態のみに依存することに着目し、オフライン環境にルートまでのパス上のノードの教材オブジェクトをコピーする方式を検討した。これらのコピーには、ポストコンディショナルルールがそのまま含まれている。各ノードは子ノードの状態をキャッシュしていて⁽¹⁵⁾⁽¹⁸⁾、子ノードの状態が変化しない限り最新状態を保持しているため、PF1外への移動が生じるまでは、オフラインでオンラインと同一の処理を継続できる。なお、サブツリーをPF1にダウンロードし、PF0に学習状態をアップロード・更新する手順も、教材オブジェクトのデザインパターンに則って規定されているが、詳細は紙幅の都合で割愛する。

7. 実用的学習支援環境への適用

これまで開発を進めてきた分散マルチプラットフォームアーキテクチャを、以前検討した学習者適応型コンテンツ⁽⁷⁾やグループ型作問学習機能⁽¹²⁾を組み合わせた学習シナリオへ適用する。図5に概要を示す。全体は、学習者適応型の独習コンテンツ、テストバンク、グループ型作問学習から構成される。まず、独習コンテンツで学習を行い、一定のレベルに達したと判断されたらテストバンクに移動する。テストバンクで理解度の詳しいチェックを行い、その結果に応じてグループ分けを行って、独習コンテンツで学習した内容を応用するグループ型作問学習を行う。共有学習目標Ob1, Ob2は、独習コンテンツからテストバンク、テストバンクから作問学習に、理解度などの学習状態を引き渡すために用いる。また、テストバンクと、それ以外の機能はPF1とPF0に分散配置され、さらに、独習コンテンツはスマートフォンにダウンロード（再配置）してオフラインで学習することも想定している。

この環境では、すべての構成要素は第4章のデザインパターンを用いて実装され、第3章の通信パターンに準拠した通信を行う。また、本稿で示したプラットフォームをまたがる共有学習目標や動的再配置方式が適用されている。テストバンクは、内部に適応型テストロジックや問題データベースを有する複雑な構成が想定されるが、教材オブジェクトとしてのインターフェースをデザインパターンに則ったものとする事で、他の教材オブジェクトと連携させることができる。

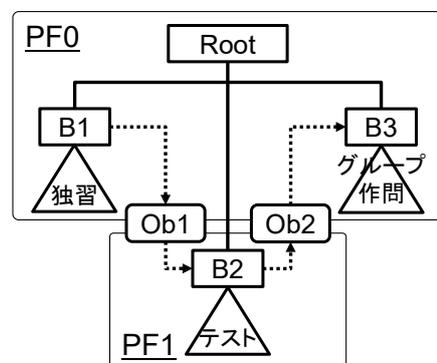


図5 実用的な学習支援環境の例

8. まとめ

拡張可能な学習支援システムアーキテクチャELECOAを基とした分散したマルチプラットフォーム

ム学習環境の検討・開発の状況について述べた。
ELECOA のデザインパターンはレイヤー型であり、物理的な通信手段を上位のレイヤーから隠ぺいすれば、さまざまな学習制御機能を実装した教材オブジェクトが複数プラットフォームで再利用可能な、分散マルチプラットフォーム学習環境を実現できる。本稿では、この特徴を生かして、教材オブジェクトをダウンロードしてオフラインで学習継続する方式、および、実用的な学習支援環境への適用例を示した。現在、これらについて最終的な実装開発を進めている。

謝辞

本研究は科研費 26280128, 17H00774 の助成を受けた。

参考文献

- (1) 仲林 清：“e ラーニング技術標準化と学習教授活動のデザイン —オープンな教育エコシステムの構築を目指して—”，人工知能学会誌, Vol.25, No.2, pp.250-258 (2010)
- (2) Advanced Distributed Learning: “Sharable Content Object Reference Model SCORM® 2004 3rd Edition” (2006)
- (3) IMS Global Learning Consortium: “IMS Question & Test Interoperability™ Specification Ver. 2.1 Final Specification” (2012)
- (4) Koper, R. and Tattersall, C. (Eds.): “Learning Design: A Handbook on Modelling and Delivering Networked Education and Training”, Springer (2005)
- (5) IMS Global Learning Consortium: “IMS Learning Design Ver. 1.0 Final Specification” (2003)
- (6) Nakabayashi, K., Morimoto, Y. and Hada, Y.: “Design and Implementation of an Extensible Learner-Adaptive Environment”, Knowledge Management & E-Learning: An International Journal, Vol.2, No.3, pp.246-259 (2010)
- (7) 仲林 清, 森本容介：“拡張性を有する適応型自己学習支援システムのためのオブジェクト指向アーキテクチャの設計と実装”，教育システム情報学会誌, Vol.29, No.2, pp.97-109 (2012)
- (8) 仲林 清, 永岡慶三：“拡張性向上のための教材オブジェクトアーキテクチャを用いた WBT システムの開発”，信学論(D-I), Vol.J88-D-I, No.6, pp.1104-1114 (2005)
- (9) 森本容介, 仲林 清, 芝崎 順司：“ELECOA における教材オブジェクト・プラットフォーム間インタフェースの設計と実装”，電子情報通信学会論文誌 D, Vol. J98-D, No.6, pp.1033-1046 (2015)
- (10) 仲林 清, 森本容介：“拡張性を有する学習支援システムアーキテクチャにおける教材オブジェクト間通信パターンの検討”，信学技報, ET2014-100 (2015)
- (11) 仲林 清, 森本容介：“拡張性を有する学習支援システムにおける教材オブジェクトのためのデザインパターンの検討と試作”，信学技報, ET2016-39 (2016)
- (12) 仲林 清, 森本容介：“拡張性を有する学習支援システムにおける再利用性向上のための教材オブジェクトデザインパターンの設計と実装”，教育システム情報学会誌, Vol.35, No.3, pp. 248-259 (2018)
- (13) 森本容介, 仲林 清, 星野忠明, 前田宏：“ELECOA を用いた cmi5 対応学習管理システムの設計と実装”，教育システム情報学会誌, Vol.37, No.1, pp.19-31. (2020)
- (14) 仲林 清, 森本容介：“拡張性を有する学習支援システムの分散マルチプラットフォーム学習環境への適用検討”，電子情報通信学会技術研究報告, ET2017-34, pp.17-22 (2017)
- (15) 仲林 清, 森本容介, 池田 満, 瀬田 和久, 田村 恭久：“拡張性を有する学習支援システムアーキテクチャに基づく分散マルチプラットフォーム学習環境の検討と試作”，教育システム情報学会研究報告, Vol.33, No.5, pp.59-66 (2019)
- (16) Buschmann, F., Meunier, R., Rohnert, H. et al.: “Pattern-Oriented Software Architecture: A System of Patterns”, John Wiley (1996)
- (17) Gamma, E., Helm, R., Johnson, R. et al.: “Design Patterns”, Addison-Wesley (1994)
- (18) 森本容介, 仲林 清：“ELECOA における教材オブジェクト間通信削減方式”，教育システム情報学会研究報告, Vol.35, No.6, 発表予定 (2021)