

OMT法による高度個別型CAIの試作

富士 隆* 谷川 健* 星原 健二郎* 伊藤佐智子* 小銭 正尚* 三枝 武男**
*SRL (学習情報通信システム研究所) **北海道情報大学

オブジェクト指向技術は、ソフトウェア開発の生産性向上に有効な手段であると注目されている。今回は、次世代CAIシステムの基盤技術開発を進める中で、オブジェクト指向分析、設計手法(OMT法)を用いて次世代CAIシステムのプロトタイプである高度個別型CAIの概念モデル構築から実装化までを行ったので、その内容、効果、そして課題について報告する。

Prototype system of Advanced and Individualized CAI with OMT

Takashi Fuji* Takeshi Tanigawa* Kenjiro Hoshihara* Sachiko Ito*
Masahisa Kozeni* Takeo Saegusa**
*Software Research Laboratory **Hokkaido Information University

Many are interested in object-oriented technology and its effectiveness to improve productivity in software development.

We are developing key technology of future CAI system. Now, we have designed conceptual design model of advanced and individualized CAI and implemented its prototype system with OMT that is one of object-oriented analysis and design methodology. We show how we have developed, what is effective and what we must solve in future in developing with OMT.

1. はじめに

次世代CAIシステムのプロトタイプである高度個別型CAIの試作を通して、学習情報に必要なデータベースの明確化とそのデータベース構築技術の研究開発を進めている。高度個別型CAIのデータベース構築要件として、

- ・マルチメディアデータと知識表現が統一的に扱える
 - ・情報の自由な探索ができる
 - ・広域ネットワーク上の分散管理にも対応できる
- 等があげられ、これらの解決手段の一つとしてオブジェクト指向技術を利用している。今回は、オブジェクト指向技術の枠組みを最大限活用することを狙ってシステム開発の上流工程であるオブジェクト指向分析、設計手法を用いて高度個別型CAIの概念モデル構築からオブジェクト指向プログラミング、オブジェクト指向データベース管理システム等を用いて実装化したのでその概要について報告する。

2. オブジェクト指向分析手法の採用理由

システム化の対象となる実世界をコンピュータ上に実現するには様々な手法があるが、ここでは、オブジェクト指向分析手法を採用した。その理由の1つは、システムの実装ではなく概念のモデル化に焦点が当てられていることである。このことにより、実装化段階で表面化する問題をモデル化の段階でチェックでき開発の生産性、品質の向上に有効である^[1]。2つめは、問題領域を「オブジェクトの集合」とそれらの間の「ネットワーク構造」としてモデル化するというオブジェクト指向の枠組み、即ち、

- ・部品という考え方（部分-全体構造）
- ・クラス継承のメカニズム（一般化-特殊化構造）
- ・メッセージのポリモーフィズム

等を利用することによりソフトウェアの再利用性と保守性で効果的^[2]と云われているからである。

オブジェクト指向分析手法にも様々な手法がある。Booch^[3]、CRC^[4]、OMT^[2]の3つの手法を比較検討した結果、分析から設計まで一貫した手法を備えているOMT法(Object Modeling Technique)を採用した。

3. OMT法

OMT法は、実世界のシステムをオブジェクトによってモデル化し、そのモデルを利用して仕様の分析から設計・実装までを支援する方法論である。OMT法では、オブジェクトモデル、動的モデル、機能モデルの3つのモデルでシステムを表現する^[2]。

オブジェクトモデルは、実世界の事態に対応したオブジェクトとオブジェクト間の関係を用いてシステムの静的な構造を記述する。機能よりもむしろオブジェクトを中心にしてシステムを開発するOMT法では、オブジェクトモデルが中心的な役割を担っている。動的モデルは、システムの状態の時間的な推移を記述したもので、状態図を含む。機能モデルは、システム内のデータ値の変換を記述したもので、データフロー図を含む。機能モデルは「何が」起きるかを、動的モデルは「いつ」、オブジェクトモデルは「何に対して」それが起きるかを仕様化する。

OMT法では、要求分析→システム分析→設計→実装の各フェーズ毎に開発方法論が規定されている。分析フェーズでは、システム分析結果を3つのモデルで表現する。設計フェーズでは、実装する環境の決定（システム設計）と3つのモデルの詳細化（オブジェクト設計）を行う。実装は、開発環境（言語、データベース）毎に指針がある。

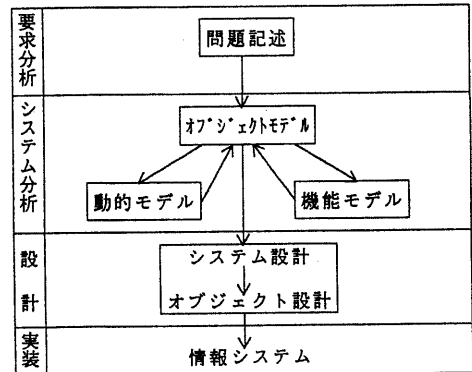


図1 OMT法によるシステム開発プロセス

4. OMT法による高度個別型CAIの分析

OMT法の手順に従って分析を実施した。

- (1) 高度個別型CAIの要件（問題記述）

今回対象としたシステムは、個々の学習者の状況に対応して教材や学習環境を提供する高度個別型CAIの試作システムである。試作の目的は、高度個別型CAIに求められるデータベースとそのデータ構造の明確化である。

現状のCAIを分析し、高度個別型CAIシステムに求められる要件を抽出した。

① 学習方法の最適化

学習者の学習目標、理解度等を把握し、最適な学習形態（チュートリアル、問題解決等）や学習の順序を決定する。

② 学習内容（説明、問題等）の自動最適化

学習履歴、理解度等より学習者に最適な教材（レベル、マルチメディア、シミュレーション等）を動的に生成する。

③ 学習者の誤り分析

問題の解答に対する誤答の原因を把握し、適切な治療を行う。

④ 学習者の学習意欲の状態把握

学習の進捗状況、キー操作のスピード変化等から学習者の学習意欲を分析し、意欲付けを行う。

⑤ 学習者の問い合わせに対応

学習者の問い合わせ（関連知識探索、質問等）に対応し、学習者主体の学習を可能にする。

これらの要件を満たすシステムの学習の流れと機能イメージを、図2に示す。

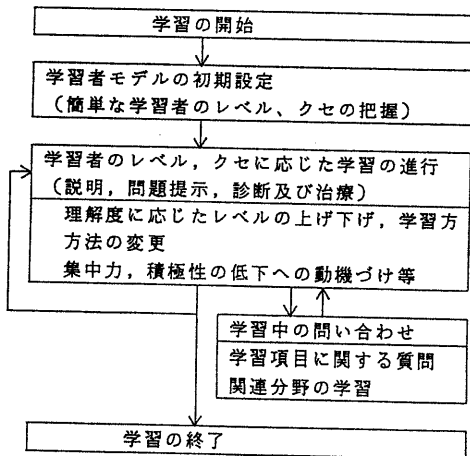


図2 学習の流れと機能イメージ

(2) 高度個別型CAIのオブジェクトモデル

OMTにおける分析とは「実世界のシステムを理解できる形で明確にモデル化すること」である。高度個別型CAIの分析作業は、現存する授業を分析するところから出発した。図3に現実世界から取り出した教室モデルを示す。

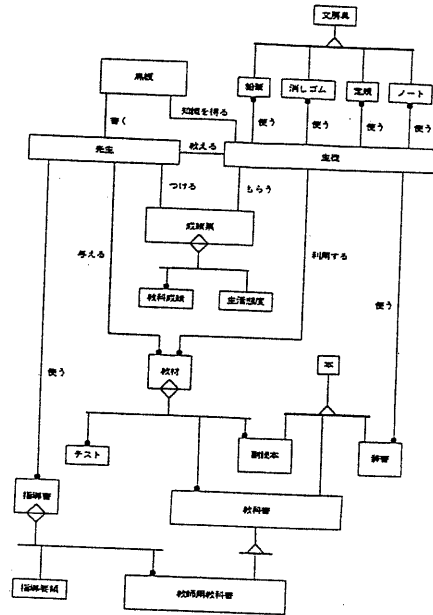


図3 初期オブジェクトモデル

初期オブジェクトモデルを洗練化するために以下のステップを実施した。

① オブジェクト及びクラスの識別

まず、高度個別型CAIの要件から名詞句を抽出しオブジェクトクラスを列挙し、その中から冗長なクラスや不適切なクラスを排除した。

② データ辞書の作成

残ったクラスについてデータ辞書を作成した。

③ オブジェクト間の集約を含む関連の識別とオブジェクト、リンクの属性の識別

ここでは高度個別型CAIの要件から動詞句を抽出し、各クラス間の関連と集約を識別した。その後、誤った関連の削除と誤った属性の削除を実施した。

④ 継承を用いたオブジェクトクラスの洗練化

このステップの目的は継承を使ってクラスを組織化させる事である。今回はボトムアップでスーパー

クラスに一般化する手法が適応できた。

オブジェクトモデルは、上記に示した作業をひとつおき実施して完成するというものではなく、他の2つのモデルを検討する中で、属性、サービス、リンクの誤り、欠落等の修正をくり返して完成する。完成したオブジェクトモデルを図4に示す。

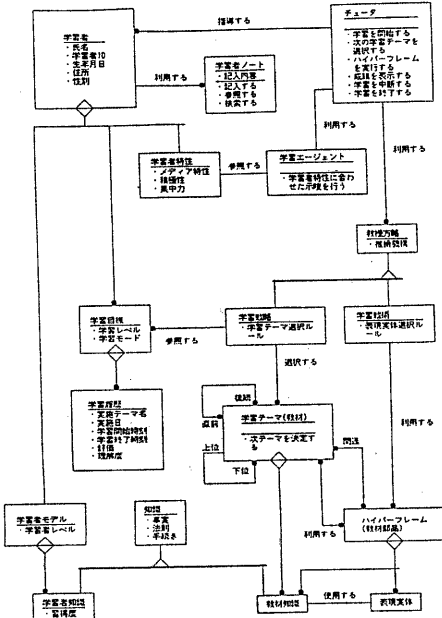


図4 高度個別型CAIのオブジェクト図

特徴のあるクラスをいくつか紹介する。

①「教授方略」クラス

学習の流れを決定する法則（学習戦略）と、提示する教材部品を選択するための法則（学習戦術）がある。学習戦略に従い、学習者の理解度に応じた学習テーマを決定する。また、学習戦術に従って、学習者の特性に合わせた教材部品を選択する（図5）。

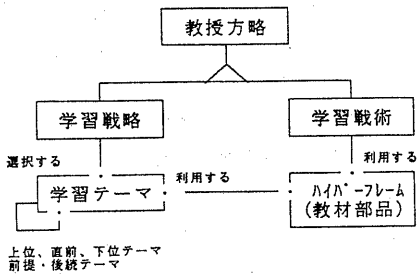


図5 教授方略のオブジェクト図

学習戦術と学習戦略は学習者の状況を判定するという共通の機能があり、この機能を上位クラスである教授方略に持つことにより、実装の効率化を図った。

②「ハイパーフレーム（教材部品）」クラス

学習の単位になるもので、テキスト、マルチメディア、シミュレーション等の表現実体と、それを支える知識を保有する。ハイパーフレームは表現実体を集約し、問題、シミュレーション及び説明は教材部品の教材名、教材種別及び学習レベル等を継承する（図6）。

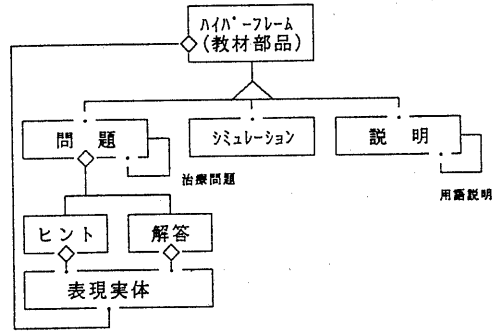


図6 教材部品のオブジェクトモデル

③「表現実体」クラス

学習者に提示する各種媒体を保持する。一つの表現実体は複数の媒体からなる。従って、表現実体がテキスト、静止画、音声、動画、アニメーション、ストーリーを集約し、それらの同期をとる機能を有する（図7）。

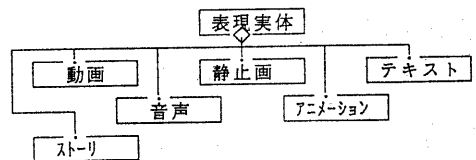


図7 表現実体のオブジェクトモデル

(3) 高度個別型CAIの動的モデル

OMT分析における次のステップは動的モデルを構築する事で、システム及びその中のオブジェクトの時間に依存した振る舞いを抽出する。ここでの目的は、オブジェクトモデルの妥当性を評価し、かつオブジェクトクラスのサービス（メソッド）を明確にすることである。このためにオブジェクト間の事象を抽出した

事象トレース図と、オブジェクト単位に状態の変化を記述した状態図を作成する。動的モデルの構築は以下の手順で実施した。

- ① 典型的な対話シーケンスのシナリオを用意する。
今回作成したシナリオの一部を図8に示す。

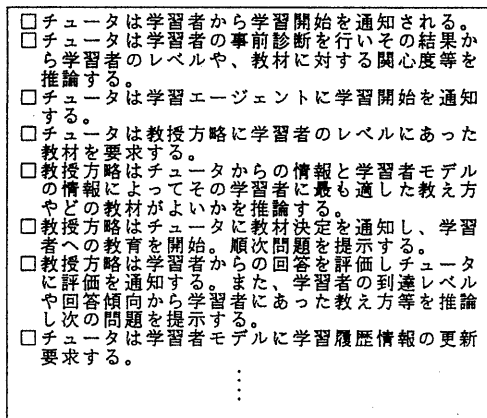


図8 高度個別型CAIのシナリオ

- ② オブジェクト間の事象を識別する。
- ③ 各々のシナリオに対して、1つの事象トレースを作成する。学習の流れの概略トレース図を図9に示す。

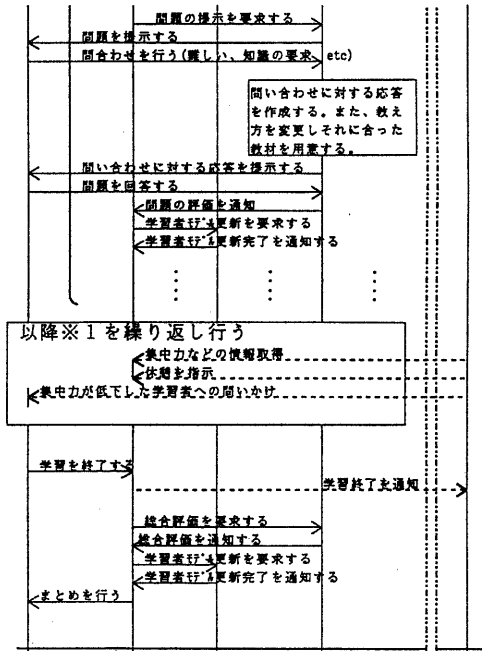
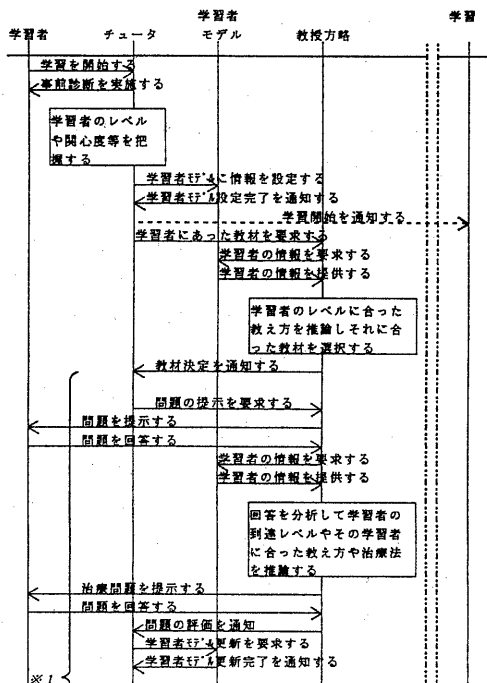


図9 学習の流れの概略

- ④ 状態図を構築する。
- ⑤ オブジェクト間の事象を対応させて一貫性を検証する。

(4) 高度個別型CAIの機能モデル

OMTによる分析の最終フェーズとして機能モデルを構築する。目的はオブジェクトモデルのオブジェクト

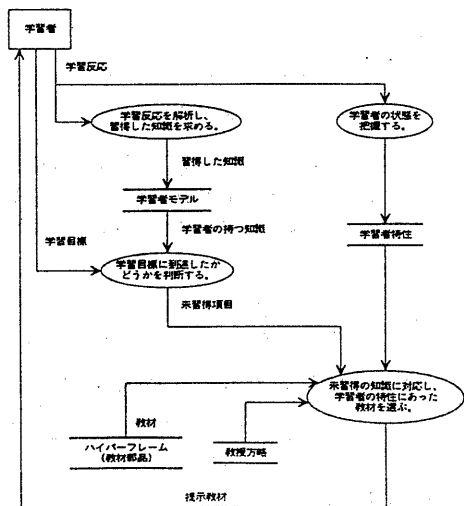


図10 学習の流れのデータフロー

トクラスや属性（データフロー）と、動的モデルの活動や動作（プロセス）を結合して、要求する機能を満足するか否かを確認することである。機能モデルの構築手順は以下のとおりである。

- ①入力値及び出力値を識別する。
- ②機能的な依存関係を表すデータフロー図を構築する。今回作成したトップレベルのデータフロー図を図10に示す。
- ③最下層のデータフロー図のプロセス毎に機能（関数）を記述する。

5. OMT法による高度個別型CAIの設計

(1) 高度個別型CAIのシステム設計

この段階ではシステムの高レベルの構造について選択する。OMT法に従って、下記のとおりシステム設計作業を進めた。

①サブシステムへの分割

OMTでは、システムの上で幾つか共通点を持っていたり、提供するサービス群によって分類されるべき要素をサブシステムと呼ぶ。システムのサブシステムへの分割は、水平方向のレイヤと垂直方向のパーティション集合として編成できる。高度個別型CAIを設計する上でもそこに主眼をおき、サブシステム分割を行った。図11にサブシステム分割作業によって得られたシステムの概略ブロック図を示す。

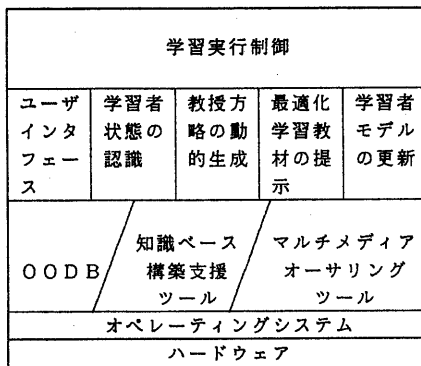


図11 概略ブロック図

②並行性の識別

今回の要件の中に、学習者の集中力や積極性の判

定と学習進行を並行して行うという項目がある。この事象は非同期であり、単一制御スレッド上にはまとめられないため並行タスクとして定義する必要がある。

前述した図9に示す学習の流れの概略の中で、学習エージェントと称しているオブジェクトがそれである。このオブジェクトはその左に位置するオブジェクト群と動きが違うのであるが、OMT記法では表現しなかったため、他オブジェクトと……線 で分離する表記にした。

その他学習の進行については、すべて同一制御スレッド上に実装できる。

③データストア方式の選択

今回は、マルチメディアデータ（静止画データ、動画データ、音声等）を教材として格納し、教材の部品化と一元管理を目指すという目標を設定した。それを実現するためには従来のリレーショナルデータベースでは不向きであるため、代替えとして何を採用するか検討を重ねた。その結果、複雑な構造を持つデータの蓄積や検索に有効な点¹⁰⁾と、オブジェクト指向開発方法論との親和性がよい点を評価し、オブジェクト指向データベースを採用した。

④ソフトウェア制御の実装方式選択

高度個別型CAIシステムではハードウェアがシングルプロセッサを採用しており、並列タスク機能は、マルチタスクOSを採用して実現した。

⑤共通アーキテクチャの枠組み

今回のシステムでは、知識ベース、マルチメディアオーサリングツール及びOODBの3つのツールを併用して実装した（詳細は6.参照）。3つのツールの複合動作を確認した事例がなかったため、3つのツールの親和性が大きな検討項目となった。その点については動作検証システムを試作し、組み合わせ実装が可能である事を確認した。

(2) 高度個別型CAIのオブジェクト設計

本フェーズの目的は分析モデルを発展させて実装に対する詳細な基盤を提供する事である。

①各操作実装用アルゴリズムの設計

今回の作業では分析フェーズで作成した3つのモデルをまとめて、最終的に高度個別型CAIに求め

られている操作を抽出し、各操作の実装用アルゴリズムを設計する事に重点をおいた。高度個別型CAIの要件定義では、「学習者個人の能力や興味の度合いによって、教材の選択方式や提示方法及び学習の進行方法を選択しそれを使用して教育する」という項目があり、この要件を実現するアルゴリズム検討を第一義とする必要があった。

具体的には、教授方略の中の学習戦略や学習戦術には相当数の時間を費やし検討を重ねた。図12にはこのフェーズでの主な検討項目を示す。

| | | | |
|-----------|--|--|--|
| 学習者に関する事項 | 学習者に対する事項 ・学習者が興味を失わないように ・学習者の集中力を高める ・学習者の学習ペースを調整する | システムへの問いかけ ・アルゴリズムの難易度を調整する ・システムが学習者の進捗を認識し、適切なアドバイスを行う | 時間管理 ・学習者の学習時間に応じて ・学習者の学習ペースを調整する |
| 問題出題 | システムから学習者への問題の出題方法、診断方法、治療方法及び評価方法 ・出題ルールの検討と決定 ・診断ルールの検討と決定 ・治療ルールの検討と決定 | 学習者の学習ペースを調整する ・学習者の学習ペースを調整する | 時間管理 ・学習者の学習時間に応じて ・学習者の学習ペースを調整する |
| 教授方略 | 学習戦略や学習戦術の各種アルゴリズムの決定 ・学習内容の個別化方式決定 ・学習モード切り替え方式決定 ・学習者のあった教材部品の選択ルールの検討と決定 | システムへの問いかけ ・アルゴリズムの難易度を調整する ・システムが学習者の進捗を認識し、適切なアドバイスを行う | 時間管理 ・学習者の学習時間に応じて ・学習者の学習ペースを調整する |

図12 オブジェクト設計における主な検討項目

②外部インタフェースの実装

高度個別型CAIの外部インタフェース実装を検討する上で最低限の条件は以下のとおりである。

- ・学習者に対して快適な操作性を提供する。
- ・文字、テキスト、静止画等の時間軸のないデータと音声や動画などの時間軸のあるデータをも扱える。かつ、その各々のソースを組み合わせて使用できる事。
- ・画面作成や教材提示用ストーリー組み込み及び取扱いが用意である事。

検討の結果、上記の条件を満たし、システム環境への親和性も併せて考慮した結果、マルチメディアオーサリングツールを実装した。これについては後述する。学習者とは全てこのツールを通してインタフェースをとる。本ツールによって、学習者に適したメディア、例えば、文章の理解が苦手な学習者には絵やグラフ及び音声を多くするなど、分りや

すい情報提供が可能となった。

6. OMT法による高度個別型CAIの実装

OMT法により設計したオブジェクトをオブジェクトの性質により3つの環境で実装した。

今回実装するオブジェクトは、以下の3つに分類できる。

- ・マルチメディアデータを扱う
- ・推論機構を要する
- ・上記以外(チュートクラス、学習者クラス等)

この3種類のオブジェクトを一つの環境で実装するためには、マルチメディアデータ処理部や推論エンジン部等を作る必要がある。今回は、生産性を考慮して、それぞれの機能を持つツールを有効利用したオブジェクトの実装を試みた。

以下の項目の実装方法をそれぞれの環境に合わせて考慮する必要がある。

- ・クラス定義
- ・オブジェクトの生成
- ・操作の呼びだし
- ・継承の利用
- ・関連の実装

(1) オブジェクト指向DBMSを用いた実装

図4に示したクラスは、オブジェクト指向DBMS(OODB)上に実装した。ただし、マルチメディアデータと推論ルールは、それぞれに適したツール上に実装した(詳細は(2),(3)を参照)。今回使用したOODBの特徴は、以下のとおりである。

- ・記述言語は、C++ベースである。
- ・オブジェクトの永続機能(DBへの保存)がある。
- ・オブジェクト間の関連(参照)を支援するクラス(Reference)があり、関連オブジェクトをアクセスする時に、そのオブジェクトがメモリにあるかディスクにあるかを意識しなくてよい。

今回使用したOODBはC++準拠であるので、OMTのオブジェクト指向言語(C++)による実装の順に従った。しかし、関連の実装については、使用したOODBが提供している参照用のクラス「Reference」を利用して実装した。Referenceクラスで関連を実装するには、関連自身をReferenceクラスに属する属性として定義する。さらに、関連を生成するメソッドと関連を参照するメソッドを用意する。この方法で、関連を定義する実体としてListクラス(OODBで提供されている)を用いると、多重度が0から複数のも

のまでを统一的に実装できる。

OODBでオブジェクトを実装した利点として、以下の点がある。

- ・オブジェクト設計の成果がスムーズに実装できる
- ・オブジェクトの永続性をシステムが管理するため、大量の実体を管理する場合には有効である。

図13にOODBによるスキーマの実装例を示す。

```
// 学習者クラス定義(student.h)
//
#ifndef _Student
#define _Student
#include <Object.h>
#include <Reference.h>
#include <List.h>
class Student : public Object
{
private:
    // 属性定義
    char* cs_name;           // 氏名
    char* cs_id;            // 学習者ID
    // 関連定義
    Reference ref_study_goal; // 学習目標
    Reference ref_study_model; // 学習者モデル
public:
    // メソッド定義
    // コンストラクタ
    Student(char* the_name=DEFAULT_STUDENT_NAME);
    Student(APL*);
    // 関連アクセス用
    void rset_study_goal(Study_goal* the_goal); // 学習目標の再設定
    Study_goal* bind_study_goal(); // 関連学習目標オブジェクト取得
    void rset_study_model(Stud_model* the_model); // 学習者モデルの再設定
    Stud_model* bind_study_model(); // 関連学習者モデルオブジェクト取得
// サービス
    void get_study_id(char* pc_id) // 学習者ID取得
    void get_study_status(char* pc_stat) // 学習者状態の取得
};
#endif
```

図13 OODBにおける実装例

(2) マルチメディアオーサリングツールを用いた実装

マルチメディアオーサリングツールとしては、以下のような機能がある。

- ・取り込んだイメージの一部のコピー、ペースト、回転、サイズの変更。
- ・静止画やテキストに対し、動画やオーディオを同期させる。
- ・ストーリーを編集して実行する。

上記のような機能を用いて、テキスト、音声、静止画像、動画像などによる学習教材の作成、提示を行った。

以下に、マルチメディアオーサリングツールを用いた学習教材の提示方法及び、マルチメディアオーサリングツールと部品化された学習教材を格納しているOODBの関係について説明する。

①学習教材の提示

学習者の理解度やレベル、メディア特性など個々の学習者に応じた教材をマルチメディアオーサリングツールで学習者に提示する。

まず、教材部品オブジェクトと説明オブジェクト内の教材種別、学習要素と説明オブジェクト内の理解度やレベルを検索して、該当する説明オブジェクトを取り出す。

次に、該当の説明オブジェクトと集約の関係にある表現実体オブジェクト内のメディア特性を検索し、該当する表現実体オブジェクトを取り出す。

最後に、該当の表現実体オブジェクトと集約の関係にあるストーリーオブジェクトからストーリーファイルの格納先を取り出す。そのストーリーをマルチメディアオーサリングツールが実行させることにより、オーディオデータ、画像データなどのファイルをアクセスして、例えば静止画像を出しながら音声を流すようなストーリーに従った内容を学習教材として学習者に提示する。

②マルチメディアデータの管理

マルチメディアオーサリングツールは、上記でも述べたようにストーリーやオーディオデータ、画像データなどのファイルを読み込んで、ストーリーに従った内容を提示する。従って、OODBにストーリー、テキスト、画像などのマルチメディアデータを持った場合、マルチメディアデータを読み込み、ファイルに書き込んで、またマルチメディアオーサリングツールでファイルを読み込むという3段階の処理を行う。この方法では、1教材

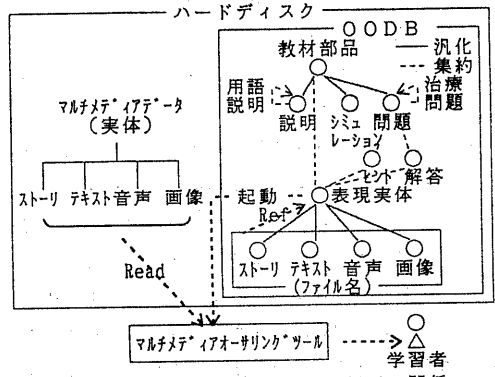


図14 マルチメディアオーサリングツールとOODBの関係

(2ページ程度)のマルチメディアデータの準備時間に10~15秒かかってしまう。そこで、あらかじめマルチメディアデータをOODBの外にファイルとして持ち、OODBの各メディアオブジェクトの中にはファイル名を持つことにより、準備時間を1秒以内におさえ、学習がスムーズに行えるようにした。

今回実装したマルチメディアオーサリングツールとOODBとの関係を図14に示す。

③ 実装上の問題点

- ・ OODBの外にマルチメディアデータを持たせることで静止画像の表示速度は半分ほどになったが、マルチメディアデータの管理方法についての問題が出てきた。
- ・ OODBではテキスト内の文字と他のメディア間に直接リンクが張れないので、用語参照を行う場合など用語と用語に対する表現実体のあて先を持つことで対応している。従ってハイパーテキストのようなテキスト内の特定の文字と他のメディア間に直接リンクが張れ、自由に探索できる構造をどう実現するかが課題である。

(3) 知識ベース構築支援ツールを用いた実装

以下のオブジェクトは推論が重要な機能となるため、推論機構をもつ知識ベース構築支援ツール上に実装した。

- ・ 教授方略
- ・ 教材知識

今回使用した知識ベース構築支援ツールは、以下のような特徴がある。

- ・ 知識表現として、フレーム構造がもてる。
- ・ 外部ルーチンとのインターフェースがあり、値の設定等が外部データベース等から容易にできる。
- ・ ルール形式の推論が記述できる。

属性としてのルールは、知識ベース構築支援ツールの機能を使って実装した。推論対象となるデータは、知識ベース構築支援ツール内のフレーム構造として定義した。ただし、実データはOODB上に管理し、OODBからツール内のフレーム構造に変換後、推論を実行させた。

例えば、学習戦術オブジェクトであれば、表現実体

選択ルールをツールで作成する。推論で使用する教材部品や学習者モデル等のデータは、OODBで管理する。教材を選択する要求がきた時に、データをツールのフレーム構造に変換し、ツールの推論エンジンを利用して、提示する表現実体を推論により決定する。

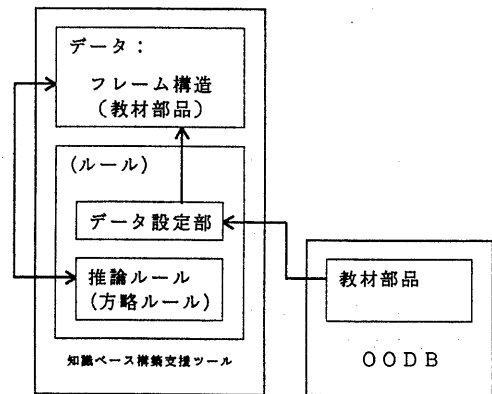


図15 知識ベース構築支援ツールにおける実装

この実装の利点は、推論エンジンを有するツールを使用することにより、生産の効率が図れることである。さらに、知識ベース構築支援ツールのデータ構造(フレーム)と、OODBのクラス構造に親和性があるため、オブジェクト設計で設計したデータ構造がそのまま実装できた。欠点は、一つのデータを二重(OODBと知識ベース構築支援ツール)で管理するため完全なカプセル化が実現できていないため、保守性の側面で課題を残している。

(4) 今後の課題

一つのシステムを複数の環境で実装したことにより、次の問題点が明らかになった。一つのオブジェクトのデータを、OODBとツールに固有な形態の2つで管理する必要があったため、完全なカプセル化を実現できなかった。

この点を解決するために、マルチメディアデータと、推論を伴う知識表現データの統合的なデータモデルを確立する必要がある。

7. OMT法の効果と課題

(1) システムライフサイクルの一貫性

一般にオブジェクト指向によるシステム開発は、システムライフサイクルの一貫性があるといわれている^[7]。今回採用したOMT法では、オブジェクトモデルを中心に分析、設計を進め、実装をオブジェクト指向で行う限りにおいては、一貫性があった。しかし、動的モデルや機能モデルをいかに利用するかは、設計者にまかされており、一定のルール化が必要であろう。

今回の開発では、OMT準拠のCASEツールを併用したが、CASEツールが3つのモデルを統合的に扱えるものではなかったため、エディタ的な使用に留まった。3つのモデルの統合化方法論が明確になり、相互チェック等が自動化されると、CASEを使用した一貫した開発が可能になると思われる。

(2) システムのモデル表現力

オブジェクトの静的な表現能力は十分あり、OODBMSのスキーマ設計等には威力を発揮した。しかし、サービス(メソッド)の表現、特に詳細な部分は、動的モデルと機能モデルに分散しており、オブジェクト単位での明確な表現方法を規定する必要がある。

また、OMT法では、オブジェクト間のメッセージ通信を表現するモデルが用意されていないので、オブジェクトの動きを一望するのに難がある。

(3) オブジェクト指向技術の枠組みの実現

オブジェクト指向の枠組みの中で、今回は「部分-全体構造」(集約)と「クラス継承のメカニズム」が有効に利用できた。

集約については教材部品が複雑な構造を持つマルチメディアデータであったが、構造をそのままオブジェクトモデルで表現することが可能であった。しかし、教材部品間の集約関係が学習者個々によりダイナミックに再編するところは、オブジェクトモデルの継承ではモデル化が不十分であった。このような動的な継承を表現することが今後の課題となる。

継承については、教授方略関連のオブジェクトに対して、共通部分の有効利用により実装時の効率化が図れた。

8. おわりに

今回の試作では、概念モデル、特にオブジェクトモデルの構築に十分時間をかけることにより不要なオブジェクト、あるいは必要なオブジェクトの確定が事前にでき実装化での問題発生 of 極小化に役立った。また、開発メンバー間の意思統一にも有効であることが確認できた。今後は、このプロトタイプをベースに関連する要素技術を組み込みながら改良していくが、オブジェクト指向の枠組みが再利用面でいかに有効に機能するのかわを確認していきたいと考えている。

謝辞

試作システムの実装化面でご協力いただいた(株)エスシーシー札幌支店、長内課長はじめ関係者の皆様に感謝致します。

参考文献

- [1]Bruegge B., Blythe J., Jackson J., Shufelt J.: "Object-Oriented System Modeling with OMT", OOPSLA'92 Conference Proceedings, pp. 359-376 (1992)
- [2]Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorenzen W.: "OBJECT-ORIENTED MODELING AND DESIGN", Prentice-Hall(1991); 羽生田栄一監訳, "オブジェクト指向方法論OMT", トップラン(1992)
- [3]Booch G.: "Object Oriented Design With Application", Benjamin/Cummings(1991)
- [4]Wirfs-Brock, Rebecca et al: "Designing Object-Oriented Software", Prince-Hall(1990)
- [5]富士隆他: "オブジェクト指向による高度個別型CAIの概念設計モデルについて", 情報処理学会研究報告93-CE-25, Vol. 93, No. 9, PP. 21-29(1993)
- [6]増永良文: "次世代データベースシステムとしてのオブジェクト指向データベースシステム", 情報処理, Vol. 32, No. 5, pp. 490-499
- [7]Martin J., Odell J. J.: "Object-Oriented Analysis & Design", Prince-Hall, pp. 67-74(1992)