

O-012

実行可能なモデルによる Web システム設計手法とその評価 A Web System Design Method Based on Executable Models and Its Evaluation

上西 司†
Tsukasa Kaminishi

北島 聡史‡
Satoshi Kitajima

小島 義幸†
Yoshiyuki Kojima

小泉 寿男‡
Hisao Koizumi

1. はじめに

近年のオブジェクト指向設計の普及により、現在、UML(Unified Model Language)に代表される標準化技術基盤が存在する。UML はシステム開発を可視化できるという利点によって広く浸透した[4]。

UML を用いれば、システムを視覚化し、開発を円滑に行うことができるが、モデルおよび、モデルによって生成されるコードの妥当性を確保することはできない。すなわち、モデルをもとに考えたコードが正しいかは実装を終えるまでは分からない。

現行のUML に実行可能なエンジンが付加した Executable UML(以下 xUML) [1]は構築するシステムで実現されるデータフローをシミュレートできる。現在 xUML は MDA(Model Driven Architecture) [2]の一部として標準化が行われている。

本稿では、xUMLを用いてシステムのシミュレーションを視覚化できるWebシステム設計手法を提案する。本方式では、実現のためにOMG(Object Management Group)で規定されているUMLアクションセマンティクス、およびそれに基づくアクション言語からなる実行環境をベースとしている[3]。本方式の評価を、ARIS(Architecture of Integrated Information System) [5]を用いたビジネスプロセスの一連の流れに組み込んだWebシステムのプロトタイプ構築、およびプロトタイプ設計手法の評価により行う。

ARIS はビジネスプロセスモデルの作成、およびビジネスプロセスモデルからの PIM(Platform Independent Model)の変換に用いる。

2. 本設計手法の実現方法

2.1 xUML を用いた設計の流れ

図1に現行のUMLを用いた設計手順を示す。各設計フェーズで適当なモデルを作成し、設計図として利用する。

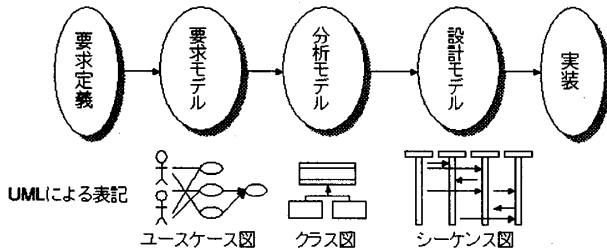


図1 WebシステムにおけるUMLを用いた設計手順

図2にxUMLを用いた設計手順を示す。システムを視覚化して開発できるというUMLの利点に加えて、作成したモデルの検証、およびシステムのシミュレーションを行うため、UMLの課題であるモデルとコードの妥当性を

†東京電機大学 理工学部 情報システム工学科

‡東京電機大学院 理工学研究科 情報システム工学専攻

実装前に確保できる。設計手順にモデルの実行とアクション言語の付加が加わるだけで、基本的な流れは変わらない。

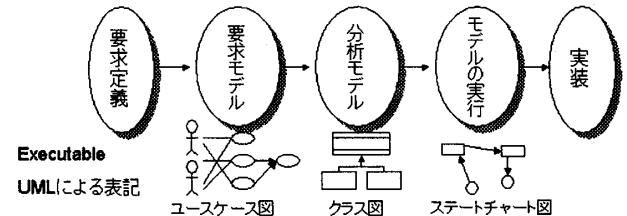


図2 WebシステムにおけるxUMLを用いた設計手順

2.2 動的シミュレーションの視覚化方式

本稿で提案する設計手法はWebシステム向けに特徴付けた手法である。以下に本稿で提案する設計手法の機能と実現方法、およびモデルから生成されるメタデータ、アクション言語、セマンティクスおよびアクション言語のコンパイルの関係を示す。

2.2.1 xUMLによるモデルの実行

以下の図はモデル実行のための一連の流れを示す。

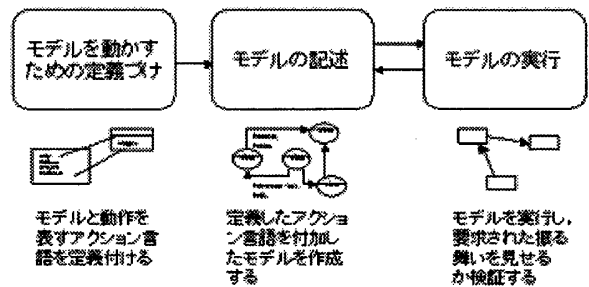


図3 モデル実行の実現

(1) モデルを動かすための定義づけ

モデルを動かすためにモデルに付加するアクション言語の作成とアクションの定義づけを行う。

アクション言語：実際にモデルに記述する言語である。アクション言語の役割はデータフローをもとにモデルできっかけとなるイベント同士を結びつけることである。

アクション言語はアクションセマンティクスに忠実でなければならない。

アクションセマンティクス：シミュレーションを行うためにはモデルに動作が詳細に記述されていなければならない。アクションセマンティクスとは、モデルに記述される動作の標準仕様である。モデルに記述される動作はシステムのデータフローによって生成される。ゆえに、シミュレーションを行うことはシステムのデータフローを追うことに似ている。

アクション言語はアクションセマンティクスに基づいて作成される。しかし、アクションセマンティクスはセマンティクスであるため、実際のアクション言語の構文などに

は触れておらず、セマンティクスさえ満たせば開発者が自由に作成できるようになっている。

(2) モデルの記述

(1) で定義したアクション言語を付加したモデルを作成する。

(3) モデルの実行

(2) で記述したモデルからアクション言語およびモデル自体の持つ情報を XMI に変換して抽出する。これらの情報と図2の要求モデル(ユースケース)から用意した入出力データを用いてコンパイラによるシミュレーションを行う。

メタモデル：モデルの内部構造にアクション言語を付加し、動作させるにはモデルにどのような意味づけがなされているかなど、モデル自体の持つ情報がやりとりできなければならない。アクション言語を含むそれらの情報をコンパイラに送り、動作させる。

本稿では意味情報を記述した UML メタモデルの記述に関する意味論や文法に MOF (Meta Object Facility) [2] を用いる。MOF はメタモデルの構成要素を定義し、またメタモデルを管理するための標準技術仕様である。

上記 MOF を基盤としてモデルの情報を XML 形式で管理するために XMI (XML Metadata Interchange) [2] を用いる。XMI は UML のメタデータとしての標準技術仕様であり、多くのベンダで採用されている。

コンパイラ：ここでのコンパイラとはモデルに記述したアクション言語を解析する機構をさす。コンパイラによってアクション言語は理解され、シミュレーションは実行される。

2. 2. 2 シミュレーションの動的表示

コンパイラによって逐次解釈されていくアクションの情報は動的にモデルにフィードバックされ、アクション言語を付加した元画面に表示される。

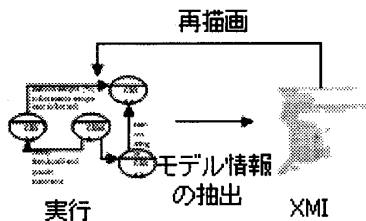


図4 動的表示の仕組み

シミュレーションにおいては 3 層構造を表現するために、モデルにクライアントの入力からデータベース(DB)への接続までの情報の記述を許し、または加え、図 4 の仕組みによってシミュレーション画面に入力とそれにとまう結果を表示させる。

3. プロトタイプ構築と評価手法

本設計手法はビジネスプロセスモデルからシステム構築に至る一連の流れの一部である。そのため、プロトタイプ構築は実際に図 5 に示すシステム開発手順に従って行う。

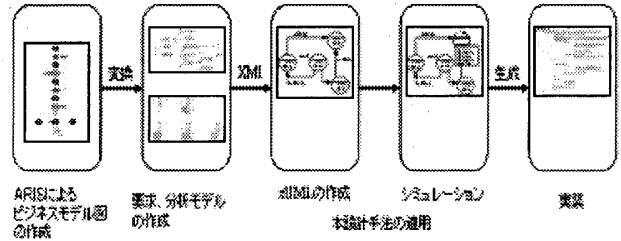


図5 チケット予約システムの概要

本設計手法を適用するのは図6に示すような3層構造のチケット予約システムである。クライアントがブラウザからアクセスし、コンサートチケットの予約を行うWebシステムを構築する。

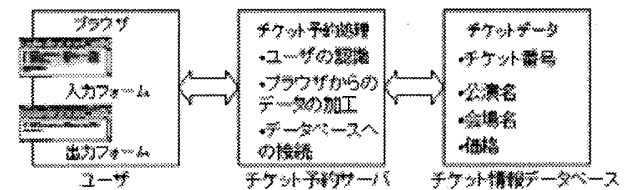


図6 チケット予約システムの概要

評価を行うために上記プロトタイプは現行の UML を用いた手法(図 1)によっても構築する。以下に評価方法を示す。

(1) 設計手法の比較

実際に、現行の UML による設計手法と我々が提案した xUML による設計手法の比較を行う。

(2) 実装段階におけるコードのステップ数による比較

構築した 2 つのプロトタイプの実装フェーズにおけるプログラムコードのステップ数を比較し、プログラミングにかかる負担を評価軸のひとつとする。

4. まとめ

xUML (実行可能なモデル) を用いてシステムのシミュレーションを視覚化できる Web システム設計手法を提案し、その実現のためのモデルを実行できる環境の構築を示した。また、ARIS を用いたビジネスプロセスの一連の流れに組み込んだ 3 層構造の Web システムのプロトタイプ構築手法、および設計手法の評価方法を示した。本設計手法および実行環境は現在構築中である。本設計手法はプロトタイプであるチケット予約システムに適用し評価する。

参考文献

[1] “Executable UML MDAモデル駆動型開発の基礎”, ステイブ・Jメラー, マークJ. バルサー, 株式会社テクノロジックアート
 [2] “MDA 導入ガイド UML を基盤としたオブジェクト指向設計・開発手法”, Anneke Kleppe, Jos Warmer, Wim Bast, 株式会社テクノロジックアート
 [3] “ActionSemantics for the UML”, Object Management Group
 [4] “UML によるオブジェクト指向開発実践ガイド”, 吉田裕之, 山本里枝子, 上原忠弘, 田中達雄, 技術評論社
 [5] “ARIS によるビジネスプロセス・モデリング”, ハインリヒ・ザイドルマイヤー, ピー・エヌ・エヌ新社