

B-19 シュレイナー・メラー分析法からプログラム変換へのアプローチ

A Proposal of the Construction Method from Object Oriented Analysis.

○川澄 明裕*

Kawasumi Akihiro

武内 慎**

Takeuchi Atsushi

古田 穂高*

Huruta Hodaka

藤本 洋**

Fujimoto Hiroshi

金子 正人**

Kaneko Masato

1. はじめに

オブジェクト分析からプログラムの実装設計する方法を提案する。今回は小規模システムの分析から実装へ変換を行った。しかし、変換規則が成り立たない場合がある。その原因を明確にし、変換ルールが成り立つようなシステム分析法を検討した。

オブジェクト指向分析法である、シュレイナー・メラー法^{[1][2]}(以下 S&M 法)を用いて分析を行い、分析結果をプログラムに実装するまでの定式化を行う。

2. 分析からプログラムへの変換の考え方

2.1 S&M 法の概要

S&M 法は情報モデル、通信モデル、状態モデルの 3 つの工程に分けて進められる。S&M 法のモデルごとの概要は表 1 のようになる。

表 1. S&M 法の概要

工程	出力
情報モデル	システムをオブジェクトごとに分類することで、システムを細かく分類する
通信モデル	時間経過に伴う、オブジェクト間の関係の振る舞いを表現する
状態モデル	オブジェクト内の動的な振る舞いを表現する

2.2 変換の考え方

S&M 法の分析結果をもとにプログラムの構成要素を決定するためにつかう各モデルとプログラムと対応付けるアルゴリズムを作る(図 1)。アルゴリズムを決めるにあたって実験を行った。

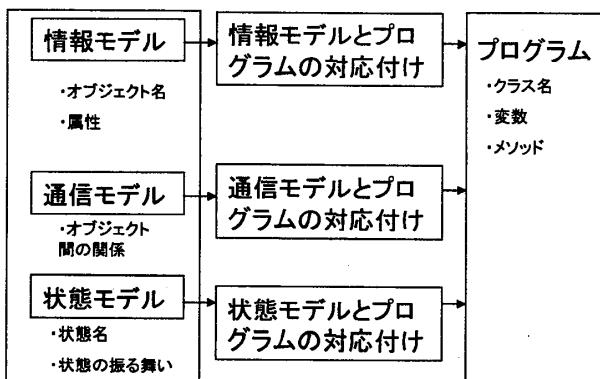


図 1. モデルとプログラムの対応

* 日本大学大学院

** 日本大学工学部

3. 実現方式と結果

3.1 対象システムの概要

OMG 日本支部が主催するロボットコンテストで用いられた、LEGO ロボットを対象システムとした。ロボットの基本動作は以下のようとする^{[3][4]}。

- 走行させるコースは白い土台に黒い線が引いてあるものである。光センサで白→黒→白→…と交互にコースを読み取ることでコース上を走るようにする
- 光センサが受け取る情報は 0~255(256 段階)である
- RCX(制御装置)はモーターを動かす
- モーターはパスファインダーを動かす
- パスファインダーが出すことのできるスピードは 0 ~7(8 段階)である

3.2 実現方式

分析モデルとプログラムの対応は以下のようになる。

(1) 情報モデル

情報モデルとプログラムの対応は図 2 の様になる。情報モデルにおけるオブジェクト名がプログラムにおけるクラス、属性が変数を表している。また属性の定義域を用い変数の型を決める。

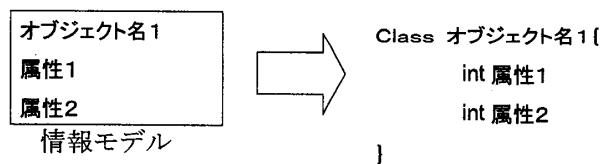


図 2. 情報モデルとプログラムとの対応

(2) 通信モデル

通信モデルとプログラムの対比は以下のようになる(図 3 参照)。

通信モデルは、クラスの呼び出しを表現している。図 3 ではオブジェクト 1 がオブジェクト 2 を呼び出し、そのクラスの機能を使うことを表現している。イベントデータ()は、呼び出すメソッド名と () 中には引数となる。これら的情報は状態モデルで用いられる。

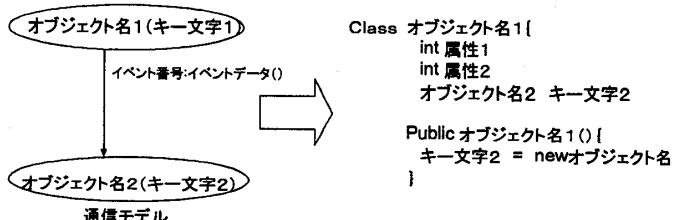


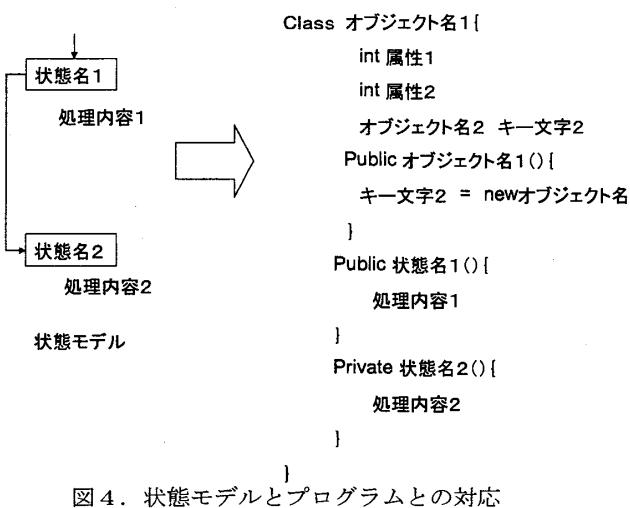
図 3. 通信モデルとプログラムとの対応

(3) 状態モデル

状態モデルとプログラムの対比は以下のようになる(図4参照)。

状態モデルにおける状態名はメソッド、振る舞いはそのメソッド内の振る舞いを表現している。メソッドの引数は通信モデルのイベントデータを参照し設定する。また通信モデルのイベントデータとメソッド名を一致させる。

状態1は外部クラスから呼び出されるためpublicとなり状態2は内部からの呼び出しなのでprivateとなる。



3.3 結果

分析結果からプログラムへ変換する際、各工程で以下の問題点が生じる。

(1) 情報モデル

プログラムでは、情報モデルで欠落した変数などが存在する。情報モデルだけでは、プログラムの実行制御するために格納する変数、繰り返し回数や、処理に必要になる変数などの抽出は困難である。

(2) 通信モデル

大きな問題は発見できなかった。

(3) 状態モデル

プログラムではif文を用いることにより、2つの状態を1つにまとめるため、分析結果とプログラムの状態数が異なる。

3.4 解決策

問題点を解決するために、各工程での解決策として以下のことが考えられる。

(1) 情報モデル

この段階で変数すべて抽出するのは困難である。このため、通信モデルや状態モデルの作成により変数を抽出して、情報モデルへフィードバックする。

(2) 通信モデル

verifier^[4](作成したモデルを動作させて、モデルの検証を行うツール)を使いシステムが動くか確認を行う。

(3) 状態モデル

if文を用いることにより2つの状態を1つにまとめられるが、変更容易性や拡張性がなくなるため、1つにまとめず、分析結果をプログラムへ変換を行う。

4. 変換プロセス

分析方法から実装までを以下のように行う(図5参照)。

(1) S&Mによる分析

S&M法によって、対象システムの分析を行う。状態モデルでは各状態が何をするかをまとめる。

(2) 情報モデルの改善

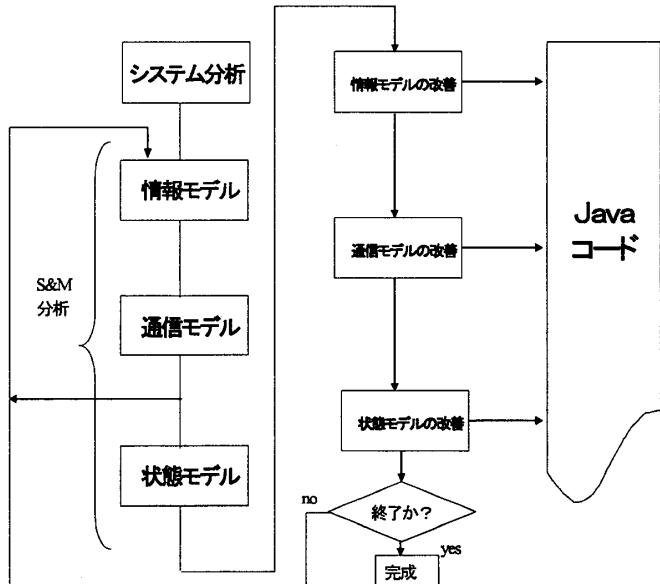
状態、通信モデルの分析により必要になった変数を情報モデルに追加する。

(3) 通信モデルの改善

verifierを行なうプログラムが動かないなどの問題が起きたら通信モデルの変更を行う。

(4) 状態モデルの改善

無駄な状態や必要な状態が抽出されたら変更し、最適化を行う。



5. 今後の課題

小規模な単純なシステムでの見通しを得た。今後は大規模システムや継承、インターフェイス、マルチスレッド等を用いるシステムへの適応を行う。

6. 謝辞

本研究を進めるに当たり、ご助言・ご協力を頂きました株式会社東陽テクニカの二上貴夫部長に感謝いたします。

7. 参考文献

[1]Sally Shlaer/Stephen. J. Mellor

「統オブジェクト指向システム分析」近代科学社 1995年

[2] レオンスター「オブジェクト・モデリング」

株式会社ピアソン/エデュケーション 1998年

[3] UML Forum-Tokyo 2002 HOME

HP:<http://www.omg.org/UMLForum2002/>

[4] 東陽テクニカ HP : <http://www.toyo.co.jp/>