

オブジェクト指向分析における 状態遷移図分割アルゴリズムの提案

北野 拓哉 山本 晋一郎 阿草 清滋
名古屋大学 工学部 情報工学科

要旨

オブジェクト指向分析の過程において、各モデル間の整合性保持のための再構築と、それに伴い複雑化していくモデル図の管理が問題としてある。特に動的モデルはクラスの仕様変更に対して変化しやすく、またオブジェクト間の通信モデルでもあるため複雑である。本稿では上記の二つの問題の解決手法として、状態遷移図分割アルゴリズムを提案する。このアルゴリズムは状態遷移図の同値類に分割された状態集合を入力とし、その各々の状態集合で新たに相互通信を行う部分状態遷移図の自動構築を行う。状態遷移図はこのアルゴリズムを適用しても分割前後の等価性が保持される。また本稿ではこのアルゴリズムの適用事例も載せる。

A Division Algorithm for State Transition Diagram on Object-Oriented Analysis

Takuya Kitano Shinichirou Yamamoto Kiyoshi Agusa
Department of Information Engineering, Nagoya University

Abstract

We propose "A Division Algorithm for State Transition Diagram" to overcome two difficulties concerning the Object Behavior Model of the Object-Oriented Analysis(OOA). One of the difficulties is to keep the consistency of the model with the other models of the OOA in the re-structuring of them. The other is to understand the model which becomes more complex by the re-structuring. The main reasons of the difficulties are that the model depends on the modification of class specifications and that it has another aspect as the communication model.

We present the algorithm and show its effectiveness by an application example.

1 はじめに

オブジェクト指向分析の基本は、実世界の中からオブジェクトを識別し、そのオブジェクトとオブジェクトどうしのつながりによって実世界を理解することである。オブジェクト指向分析では、個々のオブジェクトにシステムの果たすべき作業を分配し、オブジェクトどうしが互いに作業を依頼し合いながらシステムの機能をモデル化する。現在、オブジェクト指向分析の主流となっているアプローチはオブジェクトモデルアプローチである [1][4][5][6]。この手法は分析の初期段階で、システムの静的な構造のモデル、すなわちオブジェクトとそれを特徴付ける属性や操作、オブジェクト間の関連を示すオブジェクトモデルを構築する。オブジェクトモデルでは表現できない他の分析側面は、すなわち時間的・動作的・制御的側面は動的モデルを、変換的・関数的・機能的な側面は機能モデルを用いてそれぞれ分析を行う。

これら三つのモデルは反復的分析によって、より洗練されたモデルへ再構築されていく。この過程において二つの問題が生じる。一つ目はモデル間の整合性保持の問題である。トップダウン分割によりクラスが詳細化されてその数が増えると、まずオブジェクトモデルでクラスの全体-部分構造が構築される。そのとき他のモデルも整合性保持のためにモデルの再構築が必要となる。一般にこの整合性保持のための再構築作業は複雑である。よって、現在はこの作業の自動化や計算機支援が望まれている。

二つ目の問題は各モデルの表記図の複雑化である。各モデルはそれぞれのモデルに適合した各表記図を使用しているが、それらがモデルの詳細化にともない記述すべき内容が増え、複雑になるということである。よって、表記図に抽象-具象の階層を持たせ、抽象階層でモデルの簡潔な全体像を、具象階層でモデルの詳細部を表示する分析視点の切替え機能が必要になる。

本稿では三つのモデルのうち、動的モデルに関するこれらの問題の解決を目的とする。その解決策として状態遷移図分割アルゴリズムを提案する。まず第2章で用語を中心とした準備を行い、第3章で状態遷移図分割アルゴリズムを述べる。第4章ではこのアルゴリズムの適用によって分割された状態遷移図が、分割前と等価であることを示す。第5章ではこの状態遷移図分割アルゴリズムの適用が、上記二つの問題の解決に有効であることを示す。

2 準備

本稿で取り上げる状態遷移図は、あるクラスの状態、事象、動作、遷移を抽象化したモデルを表すものであり、状態の集合 S 、遷移の集合 Tr 、事象の集合 Eve 、動作の集合 Act 、初期状態 $is(\in S)$ 、終了状態の集合 $Fs(\subseteq S)$ の六項組で表される。遷移は非決定的である。

また Fs の要素を fs とする。状態遷移図の記法を図1に示す [4]。

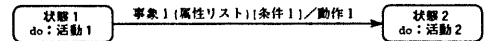


図 1: 状態遷移図の記法

状態は状態遷移図においてノードで表される。状態は図1の記法上、属性として状態名と活動を持つが、これらは状態に付随するラベルである。

遷移は状態遷移図においてアークで表される。遷移は事前状態 $pre(\in S)$ 、事後状態 $post(\in S)$ 、事象 $eve(\in E)$ 、動作 Act の四項組で表される。ある遷移 $tr(\in Tr)$ を以下の形式で表す。

$$tr(pre, post, eve, Act)$$

事象は状態遷移図において遷移のラベルで表される。事象には外部事象と内部事象の二種類ある。外部事象は対象としている状態遷移図の外部から送られる事象である。よって外部事象はその状態遷移図を用いてモデル化したクラスのインタフェースとなる。内部事象は対象とする状態遷移図内で発生する事象であり、クラスのインタフェースとはならない。ただし、外部事象でありかつ内部事象でもある事象も存在し、これは外部事象と同等に扱う。

動作は状態遷移図において遷移のラベルで表され、事象のラベルに続いて記号“/”の後に記述される。事象の集合 E と動作の集合 A との関係は $Act \subseteq Eve$ であり、動作はそれと同名ラベルの事象を発生させる。

遷移において事象と動作は空値を許す。事象が空値の場合を自動遷移と呼ぶ。自動遷移は事象ラベルのない遷移であり、事前状態の活動が完了するとただちに事後状態へ移る遷移である。事前状態に活動がない場合はその状態に入るとただちに事後状態へ遷移する。これは事前状態が初期状態のときでも同様である。ある状態から次状態への遷移が逐次的であり、その状態遷移以外に他へ影響を及ぼさない場合に、自動遷移は用いられる。

ある状態の呼び方に、次状態、前状態、隣接状態という用語を使う。ある状態の次状態とは、その状態を事前状態としているすべての遷移の事後状態の集合である。ある状態の前状態とは、その状態を事後状態としているすべての遷移の事前状態の集合である。ある状態の隣接状態とは、その状態の次状態と前状態の和集合である。

ある状態から見た遷移の呼び方に、入力遷移、出力遷移という用語を使う。ある状態の入力遷移とは、その状態を事後状態としているすべての遷移の集合である。ある状態の出力遷移とは、その状態を事前状態としているすべての遷移の集合である。

3 状態遷移図分割アルゴリズム

状態遷移図分割は、分析者によって元の状態遷移図の is と Fs 以外の状態集合を互いに素な二つの部分集合に選別される。以下では一方を状態部分集合 A 、他方を状態部分集合 B とする。 is と Fs は双方に属する。選別された状態部分集合 A と B が、以下に示す状態遷移図分割アルゴリズムに対する入力となる。アルゴリズム中の太字の用語については後で説明する。

状態遷移図分割アルゴリズム

手順1 状態遷移図を一つコピーして同じ状態遷移図を二つ作る。一方を A の状態遷移図 STD_A 、他方を B の状態遷移図 STD_B とする。以下の手順はすべて STD_A と STD_B に対して行われる。

手順2 STD_A のすべての $b \in B$ のラベルを w とする。この状態を待ち状態 (ws) といい、ラベルが w となった待ち状態 b を w_b で表す。 STD_B のすべての $a \in A$ に対しても同様の交換を行う。

手順3 すべての遷移に対して通信遷移判別を行う。その結果、通信遷移には手順4、手順5を順次に行う。

手順4 STD_A と STD_B の各通信遷移の組に対して、送信遷移と受信遷移の区別をつける。

手順5 通信遷移の組に事象と動作の作用を持たせる。これを通信遷移交換と呼ぶ。

手順6 隣接する ws に同一視を行う。

手順7 is の次状態の ws と fs の前状態の ws に対し除去可能判別を行う。

手順8 除去可能判別の結果に従い、次に挙げるパターンの遷移を除去、または交換する。ここで $*$ には任意の値が入り、空値も許す。

- 遷移 $tr(is, rws, *, *)$
- 遷移 $tr(rws, fs, *, *)$
以上の二パターンは短絡除去を行う。
- 遷移 $tr(is, nws, *, *)$
自動遷移交換を行う。
- 遷移 $tr(nws, fs, *, *)$
通信遷移交換を行う。

手順9 ws が二つ以上存在するときはそれらに同一視を行う。

状態遷移図間の事象と動作のやり取りを通信と呼ぶ。待ち状態 (ws) は、分割した状態遷移図間の通信の同期を取るために導入した状態である。 ws からの出力遷移は、分割された他方の状態遷移図で発生する内部

事象によって遷移する。また ws は手順7の除去可能判別によって除去可能な待ち状態 (rws) と除去不可能な待ち状態 (nws) の二種類に分けられる。状態遷移図間の通信の関係を変化させずに ws を除去する方法が存在すれば、それは rws となる。 rws はアルゴリズムの手順8の遷移の短絡除去によって除去される。

通信遷移判別は、状態遷移図をグラフと見なして状態をノード、遷移をエッジとし、元の状態遷移図のノードの集合を互いに素な二つの部分集合 A と B に分けるカットセットに対応する遷移を通信遷移とする手順である。 STD_A の通信遷移は以下の遷移パターンで表される。

$$tr(a, w_b, *, *) \text{ or } tr(w_b, a, *, *)$$

ここで $a \in A - \{is\} \cup Fs$ である。 STD_B に関しても同様である。

STD_A と STD_B のそれぞれの遷移は一对一の組で対応する。この遷移の組で一方が通信遷移なら他方も通信遷移となる。通信遷移には送信遷移と受信遷移の二種類ある。これらを組として扱う。送信遷移は事象を送信する遷移である。受信遷移は送信遷移から送信される事象によって遷移する。 STD_A 側が送信遷移であり、 STD_B 側が受信遷移であるならば、以下の遷移パターンで表される。

$$\text{送信遷移: } tr(a, w_b, *, *)$$

$$\text{受信遷移: } tr(w_a, b, *, *)$$

ここで $a \in A - \{is\} \cup Fs$, $b \in B - \{is\} \cup Fs$ である。 a と b を置換した場合、 STD_A 側が受信遷移、 STD_B 側が送信遷移となる。

手順5の通信遷移交換では手順4で求めた送信遷移に通信動作を加える。この動作は通信遷移の組の数だけ存在し、追加した動作の集合を Act' とする。よって、事象集合 Eve は $Eve \cup Act'$ に、動作集合 Act は $Act \cup Act'$ にそれぞれ拡張される。受信遷移の事象ラベルを、対応する送信遷移の動作ラベルにし、受信遷移の遷移条件として $[]$ で囲った送信遷移の事前状態を事象ラベルの直後に付ける¹。これにより、通信遷移は共有事象を介した STD_A と STD_B の間の相互作用を表す。

手順1から手順5までの交換規則を以下にまとめる。

$$\begin{aligned} & tr(a, b, eve_1, *) \\ & \Downarrow \text{手順1} \sim \text{2(分割)} \\ & \begin{cases} tr(a, w_b, eve_1, *) & : STD_A \\ tr(w_a, b, eve_1, *) & : STD_B \end{cases} \\ & \Downarrow \text{手順3} \sim \text{5(通信遷移交換)} \\ & \begin{cases} tr(a, w_b, eve_1, act_1) & : \text{送信遷移} \\ tr(w_a, b, act_1[a], *) & : \text{受信遷移} \end{cases} \end{aligned}$$

¹ 分析者はアルゴリズム終了後、これらのラベルを適切な名前に変更し直す必要がある。

手順6は隣接する ws に対して同一視を行う。すべての ws は他のオブジェクトからの事象の送信を待つという同一の状態を表し、これらを同一視する。ただし ws の同一視により、隣接する ws の間の自己遷移を含む遷移を除去する。自己遷移とは、事前状態と事後状態が同一の状態である遷移のことである。遷移が除去されるまでの過程を以下にまとめる。

$$\begin{aligned} & tr(a_1, a_2, eve_1, *) \\ & \quad \Downarrow \text{手順1} \sim 2(\text{分割}) \\ & \left\{ \begin{array}{l} tr(a_1, a_2, eve_1, *) : STD_A \\ tr(w_{a_1}, w_{a_2}, eve_1, *) : STD_B \end{array} \right. \\ & \quad \Downarrow \text{手順6}(w_{a_1} \text{ と } w_{a_2} \text{ の同一視}) \\ & tr(a_1, a_2, eve_1, *) \end{aligned}$$

ここで $a_1, a_2 \in A$ である。 B に関しても同様である。手順6の処理で除去されるのは $tr(ws, ws, *, *)$ だけであり、他の遷移や状態が除去されたり変換されたりすることはない。この手順6の実行後、残された ws はすべて隣接しない。よって、手順9の ws の同一視では遷移の除去は起こらない。手順9まで行くと、各状態遷移図の ws の数は高々一つとなる。

手順7の除去可能判別は隣接する ws に対して待ち状態同一視を行ったことが前提であり、 is の次状態の ws と fs の前状態の ws に対してのみ適用する。

以下の条件を満たす ws を nws とする。

- 遷移 $tr(is, ws, *, *)$ の ws であり、かつ ws の入力遷移に通信遷移が存在する。
- 遷移 $tr(ws, fs, *, *)$ の ws であり、かつ ws の出力遷移に通信遷移が存在する。

nws でない ws を rws とする。

手順8は場合分けによって、それぞれの遷移に対して短絡除去、自動遷移変換、通信遷移変換を行う。

短絡除去は遷移 $tr(is, rws, *, *)$ と遷移 $tr(rws, fs, *, *)$ に対して行う。この tr の除去には状態の同一視と他の遷移の変換がともなう。 $tr(is, rws, *, *)$ については、 rws を is と同一視して rws の出力遷移の事前状態を is に書換え、 tr を除去する。 $tr(rws, fs, *, *)$ については、 rws を fs と同一視して rws の入力遷移の事後状態を fs に書換え、 tr を除去する。

自動遷移変換は遷移 $tr(is, nws, *, *)$ に対して行う。 tr を自動遷移 $tr(is, nws, -, -)$ に変換する。ここで $-$ は空値を表す。

通信遷移変換は遷移 $tr(nws, fs, *, *)$ に対して行う。 tr を受信遷移とし、その遷移の組となる他方の状態遷移図の遷移を送信遷移とする。変換の手順は手順5と同じである。

4 分割前後の等価性の証明

3章で示したアルゴリズムは、双模倣に基づく等価性を保存する。この章では等価性の定義と、アルゴリズム等価性保存の性質を持つことを証明する。

4.1 状態遷移図の等価性の定義

トレースはある一つの外部事象系列である。トレースの実行途中に、内部事象が非同期に発生し、これによって状態が遷移する場合がある。この内部事象はクラスのインタフェースではなく、クラスの外部からは見えない。あるトレース $t (= eve_1 \cdots eve_n \in Eve^*, n \geq 0)$ の実行によって、ある状態 $p(\in S)$ からある状態 $q(\in S) \rightarrow$ 遷移することを $p \xrightarrow{t} q$ で表す。

状態集合 S_1 と S_2 があり、状態 $p(\in S_1)$ と $q(\in S_2)$ が関係 R にあることを $p R q$ で表す。状態の関係 R が双模倣であるとは、 $p R q$ ならば、任意のトレース $t \in Eve^*$ について以下の二つの条件が成り立つことである。

- $p \xrightarrow{t} p'$ ならば、 $\exists q'. (q \xrightarrow{t} q', p' R q')$
- $q \xrightarrow{t} q'$ ならば、 $\exists p'. (p \xrightarrow{t} p', p' R q')$

二つの状態遷移図 STD_1 と STD_2 が等価であるとは、 STD_1 の状態集合 S_1 と STD_2 の状態集合 S_2 の間に以下の関係 $R(S_1^I \times S_2^I)$ を定義し、この関係 R に双模倣が成り立つことである。

$$\begin{aligned} & s_1 R s_2 \text{ s.t.} \\ & \forall p \in s_1, \exists p' \in s_2 \text{ かつ } \forall q \in s_2, \exists q' \in s_1 \end{aligned}$$

4.2 等価性の証明

分割前の状態遷移図を STD_1 、 STD_1 の互いに素な二つの状態部分集合をそれぞれ A と B 、分割後の状態遷移図を STD_2 とする。

分割アルゴリズムの手順1から手順5までの実行前後における等価性を証明する。ここまで手順における遷移の変換は、以下の変換規則で表される。ただし $a \in A$ 、 $b \in B$ である。この変換規則を STD_1 に適用する。

$$\begin{aligned} & STD_1 \quad tr(a, b, eve_1, *) \\ & \quad \Downarrow \text{手順1} \sim 5(\text{分割と通信遷移変換}) \end{aligned}$$

$$STD_2 \quad \left\{ \begin{array}{ll} tr(a, w_b, eve_1, act_1) & : \text{送信遷移} \\ tr(w_a, b, act_1[a], *) & : \text{受信遷移} \end{array} \right.$$

ここで STD_2 に二つの状態遷移図ができる。

STD_1 の状態集合 S_1 と STD_2 の状態集合 S_2 の関係 R は以下ようになる。

$$R = \{(a, \{a, w_a\}), (b, \{w_b, b\})\}$$

この関係 R に双模倣が成り立つことを証明する。

$(a, \{a, w_a\}) \in R$ が成り立つ。 $(b, \{w_b, b\}) \in R$ も成り立つ。 STD_1 の状態はトレース $t = eve_1$ によつ

て a から b へ遷移する。ここで同じトレース t によって STD_2 の状態が $\{a, w_a\}$ から $\{w_b, b\}$ へ遷移することを示せばよい。

STD_2 では新しい内部事象 act_1 が加えられ、かつその act_1 は eve_1 の発生の後に発生する。この事象系列 $eve_1 \cdot act_1$ の実行により STD_2 は以下のような状態遷移を起こす。

$$\{a, w_a\} \xrightarrow{eve_1} \{w_b, w_a\} \xrightarrow{act_1} \{w_b, b\}$$

act_1 は内部事象なので、この状態遷移を引き起こすトレースは eve_1 となる。このトレースは t と等しくなる。以上で、 STD_1 と STD_2 の状態間の双模倣のうち一方の条件が成り立つ。逆方向の模倣も明らかに成り立つ。よって、関係 R には双模倣が成り立ち、 STD_1 と STD_2 の間の等価性が成立する。

手順 6 の隣接する ws に対する同一視の実行前後における等価性の証明を行う。手順 6 における遷移の変換は、以下の変換規則で表される。ただし $a_1, a_2 \in A$ である。この変換規則を STD_1 に適用する。(注: STD_2' は STD_2 への変換途中過程である)

$$\begin{aligned} STD_1 & \quad tr(a_1, a_2, eve_1, *) \\ & \quad \Downarrow \text{手順 1} \sim 2 (\text{分割}) \\ STD_2' & \quad \begin{cases} tr(a_1, a_2, eve_1, *) & : STD_A \\ tr(w_{a_1}, w_{a_2}, eve_1, *) & : STD_B \end{cases} \\ & \quad \Downarrow \text{手順 6} (w_{a_1} \text{ と } w_{a_2} \text{ の同一視}) \\ STD_2 & \quad tr(a_1, a_2, eve_1, *) : STD_A \end{aligned}$$

STD_1 の状態集合 S_1 と STD_2 の状態集合 S_2 の関係 R は以下ようになる。

$$R = \{(a_1, \{a_1, ws\}), (a_2, \{a_2, ws\})\}$$

ここで S_2 の ws は w_{a_1} と w_{a_2} を同一視した待ち状態であり、この状態を w_{a_1} としても w_{a_2} としても良い。よって、 ws は STD_1 の a_1 と a_2 の両方と関係 R の条件が成立する。

この関係 R は明らかに双模倣が成り立つ。よって、 STD_1 と STD_2 の間の等価性が成立する。

a_1, a_2 を $b_1, b_2 (\in B)$ に置換しても同じである。

手順 7 は除去可能判別を行うだけであり、状態遷移図の構成要素は変化しない。

手順 8 の短絡除去の実行前後における等価性の証明を行う。短絡除去のうち、状態の同一視の部分は手順 6 の同一視と同じである。すなわち遷移

$tr(is, rws, *, *)$ は事後状態の rws を事前状態の is と同一視して tr を除去し、遷移 $tr(rws, fs, *, *)$ は事前状態の rws を事後状態の fs と同一視して tr を除去する。よって、これらの遷移についての等価性は手順 6 の ws の同一視と同様に成立する。

遷移 $tr(is, rws, *, *)$ の除去にともなう他の遷移の変換は、以下の変換規則で表される。ただし $a \in A, b \in B$ である。この変換規則を STD_1 に適用する。

$$\begin{aligned} STD_1 & \quad tr(a, b, eve_1, *) \\ & \quad \Downarrow \text{手順 1} \sim 5 (\text{分割と通信遷移変換}) \\ STD_2' & \quad \begin{cases} tr(a, w_b, eve_1, act_1) & : \text{送信遷移} \\ tr(w_a, b, act_1[a], *) & : \text{受信遷移} \end{cases} \\ & \quad \Downarrow \text{手順 7} \sim 8 (\text{短絡除去}) \\ STD_2 & \quad \begin{cases} tr(a, w_b, eve_1, act_1) \\ tr(is, b, act_1[a], *) & : w_a \text{ は } is \text{ と同一視} \end{cases} \end{aligned}$$

STD_1 の状態集合 S_1 と STD_2 の状態集合 S_2 の関係 R は以下ようになる。

$$R = \{(a, \{a, is\}), (b, \{w_b, b\})\}$$

STD_2' の w_a は STD_1 の a と関係 R の条件が成立する。 S_2 の is は w_a に同一視された状態である。よって、 STD_2 の is は STD_1 の a と関係 R の条件が成立する。

この関係 R に双模倣が成り立つことを証明する。

$(a, \{a, is\}) \in R$ が成り立つ。 $(b, \{w_b, b\}) \in R$ も成り立つ。 STD_1 の状態はトレース $t = eve_1$ によって a から b へ遷移する。ここで同じトレース t によって STD_2 の状態が $\{a, is\}$ から $\{w_b, b\}$ へ遷移することを示せばよいが、これは手順 1 から手順 5 までの実行前後における等価性の証明の所と同様に成り立つ。以上で、 STD_1 と STD_2 の状態間の双模倣のうち一方の条件が成り立つ。逆方向の模倣も明らかに成り立つ。よって、関係 R には双模倣が成り立ち、 STD_1 と STD_2 の間の等価性が成立する。

遷移 $tr(rws, fs, *, *)$ の除去にともなう他の遷移の変換についても同様に等価性が成立する。

手順 8 の自動遷移変換の実行前後における等価性の証明を行う。自動遷移変換は、以下の変換規則で表される。ただし $a \in A$ である。この変換規則を STD_1 に適用する。

$$\begin{aligned} STD_1 & \quad tr(is, a, eve_1, *) \\ & \quad \Downarrow \text{手順 1, 2, 7, 8} (\text{分割と自動遷移変換}) \\ STD_2 & \quad \begin{cases} tr(is, a, eve_1, *) & : STD_A \\ tr(is, nws, -, -) & : STD_B \end{cases} \end{aligned}$$

STD_1 の状態集合 S_1 と STD_2 の状態集合 S_2 の関係 R は以下ようになる。

$$R = \{(is, \{is, is\}), (a, \{a, nws\})\}$$

この関係 R は明らかに双模倣が成り立つ。よって、 STD_1 と STD_2 の間の等価性が成立する。

a を $b (\in B)$ に置換しても同じである。

手順 8 の通信遷移変換は遷移の事後状態が fs であるだけで、本質的に手順 5 と同じである。よって、この変換前後の等価性は成立する。

手順 9 の待ち状態同一視は、遷移の除去が行なわれない同一視である。手順 9 の実行前後で STD_2 のすべての遷移は変化しない。また二つ以上の ws は同一視されるが、手順 6 の同一視と同様、同一視後の ws は

同一視前のどの ws と同じ状態と見なして良い。よって、手順6で定義した関係 R と同様に、 STD_2 の ws は STD_1 の各状態と関係 R の条件が成立する。

関係 R が定義されたら、この R に双模倣が成り立つのは明らかである。よって、 STD_1 と STD_2 の間の等価性が成立する。

以上の証明により、手順1から手順9までの各手順を実行しても、 STD_1 と STD_2 の間の等価性は保持されることが証明される。

4.3 状態遷移図分割の例

図2を STD_1 とする。以下では、アルゴリズムを適用した STD_1 の分割例を示す。

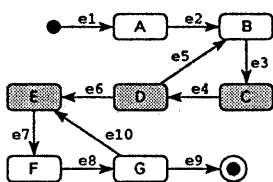


図2: 分割前の状態遷移図 STD_1

STD_1 の状態集合 S_1 と外部事象集合 E_1 は

$$S_1 = \{A, B, C, D, E, F, G\}$$

$$E_1 = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$$

である。

まず状態集合 S を二分する。図2の白色の状態を状態集合 S_A 、灰色の状態を状態集合 S_B とする。すなわち S_1 は以下の互いに素な二つの状態部分集合 S_A と S_B に分解される。

$$S_A = \{A, B, F, G\}$$

$$S_B = \{C, D, E\}$$

この二つの状態部分集合 S_A と S_B をアルゴリズムの入力として STD_1 を分割していくと、最終的に図3の S_A の状態遷移図 STD_A と4の S_B の状態遷移図 STD_B が得られる。

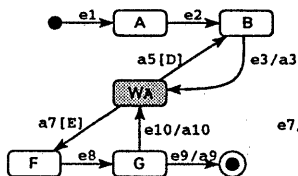


図3: 状態遷移図 STD_A

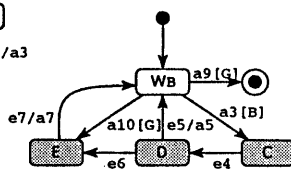


図4: 状態遷移図 STD_B

ここで S_A と S_B は

$$S_A = \{A, B, F, G, WS_A\}$$

$$S_B = \{C, D, E, WS_B\}$$

となる。また STD_A と STD_B のそれぞれの外部事象集合 E_A と E_B は

$$E_A = \{e_1, e_2, e_3, e_8, e_9, e_{10}\}$$

$$E_B = \{e_4, e_5, e_6, e_7\}$$

となり、 $E_A \cup E_B = E$, $E_A \cap E_B = \emptyset$ となる。これはクラスのインタフェースも互いに素な部分集合に分割されたことを意味する。

STD_A と STD_B を合わせて STD_2 とし、 STD_1 と STD_2 の等価性を証明する。

STD_1 の状態集合 S_1 と STD_2 の状態集合 $S_A \cup S_B$ の関係 R は以下ようになる。

$$R = \{(is, \{is, is\}), (A, \{A, WB\}), (B, \{B, WB\}), (C, \{WA, C\}), (D, \{WA, D\}), (E, \{WA, E\}), (F, \{F, WB\}), (G, \{G, WB\}), (fs, \{fs, fs\}), \}$$

トレースを $t = e_1 \cdot e_2 \cdot e_3 \cdot e_4 \cdot e_5 \cdot e_6 \cdot e_7 \cdot e_8 \cdot e_9 \cdot e_{10}$ とする。この t の任意の部分事象系列について、関係 R に双模倣が成り立つ。よって、 STD_1 と STD_2 の間の等価性が成立する。

5 事例研究

状態遷移図分割アルゴリズムを用いた動的モデルの再構築、および抽象化階層の構築を、事例「自動販売機」を実際に分析することによって説明する。

5.1 自動販売機の要求仕様書

客は自動販売機に対して硬貨を投入する。自動販売機は投入された硬貨の金額を累積計算し、その金額がある商品の値段以上になったらその商品ボタンを点灯させる。客は点灯した商品ボタンを押せばその商品を手に入れることができるが、点灯していない商品のボタンを押しても、自動販売機は何も応答しない。また売り切れの商品のボタンは常に売り切れランプを点灯させておき、この商品を買えないことを表示する。自動販売機は押されたボタンの商品を出し、この時点でのおつりを計算する。つり銭計算の結果、この金額がまだ他の商品の値段以上であったなら、その商品ボタンを点灯させる。そうでなかったら、つり銭を返却する。またこの自動販売機には返却ボタンがあり、このボタンが押されたなら、合計金額をリセットしてその金額を返却する。

5.2 システムの動的モデルの構築

前述の要求仕様書や専門家、システムのユーザに対するインタビューを元に、システムのシナリオの作成する(紙面の都合上省く)。システムのシナリオを元に

構築したシステムの動的モデル (状態遷移図) を図5に示す。

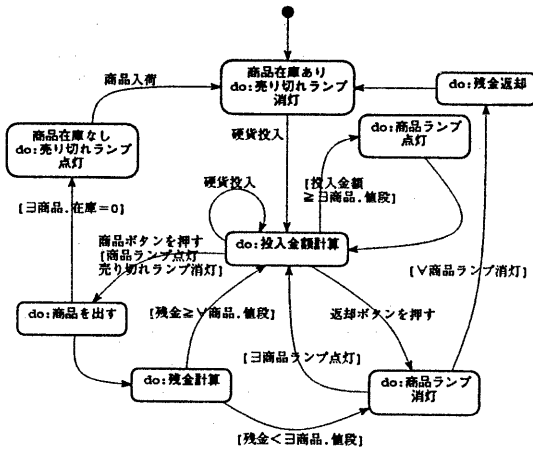


図 5: 自動販売機の状態遷移図

図5は一つのクラス“自動販売機”の状態遷移図であるが、自動販売機は図6に示すように、部分クラス“商品ボタン”、“計算機”、“本体”の三つのクラスからなる集約クラスであり、かつクラス“商品ボタン”は、部分クラス“商品ランプ”、“売り切れランプ”からなる集約クラスであると識別したとする。

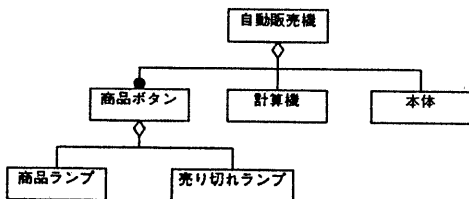


図 6: 自動販売機のオブジェクトモデル

図5の状態遷移図を図6のオブジェクトモデルの各クラスに対応する動的モデル (状態遷移図の集合) とするために、状態遷移図分割アルゴリズムを用いて状態遷移図の全体-部分構造を構築する。図5の状態のうち、状態“do: 商品ランプ点灯”、“do: 商品ランプ消灯”、“商品在庫あり”、“商品在庫なし”をクラス“商品ボタン”が有する状態として識別し、これを選択する。分割した結果が図7、図8である。なお灰色に網掛けしてある事象や動作は、通信遷移変換で新たに加えたそれぞれのラベルである。

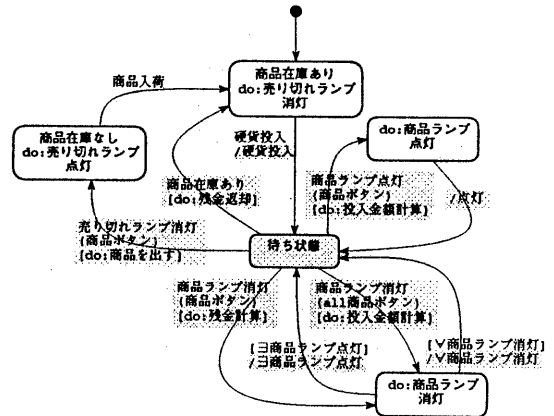


図 7: クラス“商品ボタン”の状態遷移図

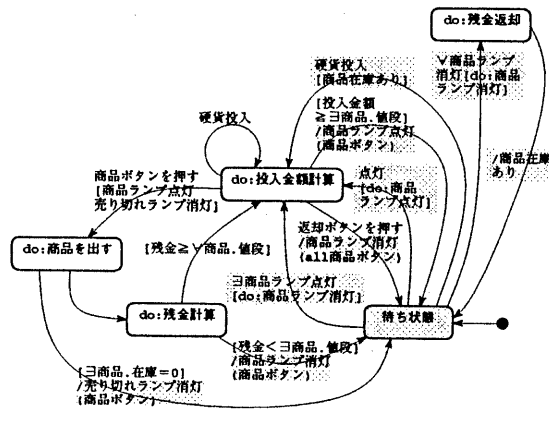


図 8: 図6を取り除いた図4の差分の状態遷移図

図8の状態のうち、状態“do: 投入金額計算”、“do: 残金計算”をクラス“計算機”が有する状態として識別し、これを選択する。分割した結果が図9、図10である。

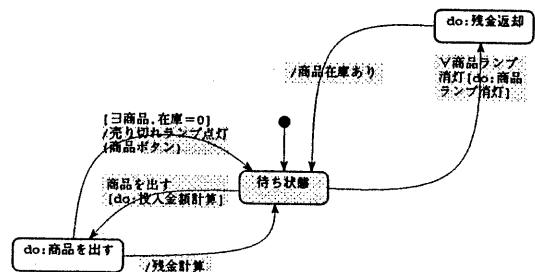


図 9: クラス“本体”の状態遷移図

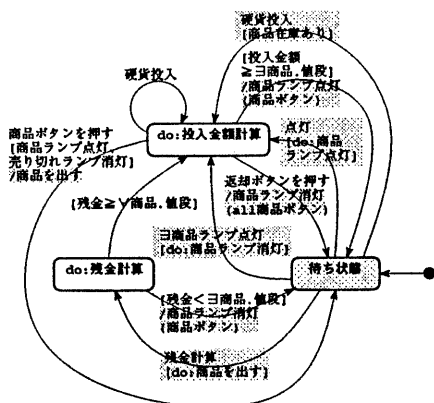


図 10: クラス“計算機”の状態遷移図

図 7 の状態のうち、状態“do: 商品ランプ点灯”、“do: 商品ランプ消灯”をクラス“商品ランプ”が有する状態として識別し、これを選択する。分割した結果が図 11、図 12 である。

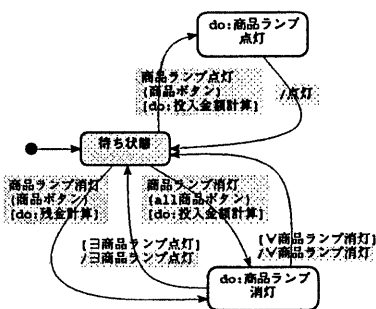


図 11: クラス“商品ランプ”の状態遷移図

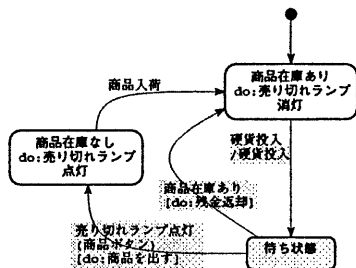


図 12: クラス“売り切れランプ”の状態遷移図

以上で図 9、図 10、図 11、図 12 の状態遷移図が、図 6 のオブジェクトモデルのクラス“本体”、“計算機”、“商品ランプ”、“売り切れランプ”のそれぞれの状態遷移図となる。クラス“自動販売機”の全体-部分の構

造化による動的モデルの再構築が、状態遷移図分割アルゴリズムの適用によって自動的になされた。またこれは同時に動的モデルの抽象-具象階層も構築されている。例えば図 7 は図 5 の抽象状態遷移図と見なせる。図 7 はクラス“自動販売機”の計算機部分と本体部分の状態を一つの状態にまとめ、商品ボタンの状態のみを着目した状態遷移図である。すなわち、状態遷移図の分析視点の切替えを行っている。

6 おわりに

分析の過程における詳細化にともなうクラスの数の増大に関して問題となる、モデルの再構築、モデルの表記図の複雑化についての解決案を提案した。すなわち状態遷移図分割アルゴリズムを用いた動的モデル構築支援法の提案である。これは状態遷移図を分割することにより、動的モデルに全体-部分構造を持たせるだけでなく、複雑な状態遷移図を抽象化し、大規模モデルの全体像を表すモデルを構築するというモデルの視点切替えにも適用可能であることも示した。また状態遷移図分割アルゴリズムはグラフ理論をベースにしているので、そのような表記図を用いてモデル化する他のモデル、例えばデータフローダイアグラムなどの分割、抽象化にも十分応用が効くと推測する。今後はデータフローダイアグラムの構築支援についても考え、オブジェクト指向分析の三つのモデル全体の構築支援及び管理を行うツールの作成を目標に取り組みたい。

参考文献

- [1] Peter Coad and Edward Yourdon, “Object-Oriented Analysis, Second Edition,” *Prentice-Hall, Inc.*, 1991.
- [2] David W. Embley, Barry D. Kurtz, Scott N. Woodfield, “Object-Oriented Systems Analysis, A Model-Driven Approach,” *Prentice-Hall, Inc.*, 1992.
- [3] Kenneth S. Rubin and Adele Goldberg, “Object Behavior Analysis,” *Communication of the ACM*, Vol.35, No.9, 1992.
- [4] James Rumbaugh, et al., “Object-Oriented Modeling and Design,” *Prentice-Hall, Inc.*, 1991.
- [5] Sally Shlaer and Stephen J. Mellor “Object-Oriented Systems Analysis,” *Prentice-Hall, Inc.*, 1988.
- [6] Sally Shlaer and Stephen J. Mellor “Object Life-cycles,” *Prentice-Hall, Inc.*, 1992.
- [7] 富樫 敦, “プロセス代数と等価性 (前編),” *bit*, Vol.23, No.12, 共立出版, 1991.