

ペトリネットによるオブジェクトマイグレーション振舞いモデリング

佐藤秀樹^{*} 林達也^{**}

^{*}株式会社デンソー

^{**}名古屋工業大学電気情報工学科

実世界の実体の多面的な性質の表現, 任意の側面 (aspect) の動的な獲得・喪失を扱う多面的データモデルに基づくデータベースでは, オブジェクトの性質の進化に伴い, オブジェクトのクラスへの所属関係の更新であるオブジェクトマイグレーション (Object Migration; OM) が起きる. 本論文では, 多面的オブジェクト指向データモデル MAORI (Multi-Aspects Object-Oriented Data Model) に基づくデータベースにおいて, ペトリネット応用による OM 振舞いモデルの構築を提案する. モデル化のために述語/遷移ネットを拡張したクラス/遷移ネットを提案し, これに基づき, 遷移制約に対する OM 振舞い構造ネット, 静的制約に対する OM 振舞いチェックネットを構築する.

Object Migration Behavior Modeling with Petri-Nets

Hideki Sato^{*} Tatsuya Hayashi^{**}

^{*}DENSO CORPORATION

^{**}Dept. of Electrical & Computer Engineering, Nagoya Institute of Technology

In databases based on a multi-aspects object-oriented data model which aims to represent multiple aspects of real-world entities and to handle dynamical acquiring/losing of these aspects, object migration (hereafter OM) occurs, as the properties of an object evolve in its lifetime. This paper proposes OM behavior model for databases based on MAORI, a multi-aspects object-oriented data model. In the modeling framework, Class/Transition net, an extension of Predicate/Transition net, is used to make OM Behavior Structure net for transition constraints and OM Behavior Check net for static constraints.

1. はじめに

オブジェクトの多面的な性質の表現, 任意の側面(aspect)の動的な獲得・喪失を扱う多面的データモデルの研究[1]-[5]も行われてきた。多面的データモデルでは, (1)オブジェクトの属性値の更新だけでなく, (2)オブジェクトのクラスへの所属関係の更新においても, オブジェクト識別性(object identity)が維持される。(1)に関する更新操作の体系はオブジェクトの振舞い(behavior)とよばれ, オブジェクトの一貫性を維持するために, そのモデル化および設計技法の研究が多数行われてきた[6]-[11]。一方, (2)の更新はオブジェクトマイグレーション(Object Migration; OM)とよばれ, オブジェクトの生存期間においてその性質が進化することに伴って起きる。OMは(1)の更新と同様にオブジェクトの一貫性を維持するうえで重要である。しかし, OMに関する更新操作の体系, すなわちOMの振舞いのモデル化および設計技法の研究は, 未だ十分には行われていない。

本論文では, 著者等が提案した多面的オブジェクト指向データモデルMAORI (Multi-Aspect Object-Oriented Data Model) [12]を前提として, ベトリネット[13]の応用によるOMの振舞いモデルの構築を提案する。この振舞いモデルは, OM振舞い構造(Object Migration/Behavior Structure; OM/BS)サブモデルとOM振舞いチェック(Object Migration/Behavior Check; OM/BC)サブモデルとから成る。OM/BSサブモデルはOMの振舞い構造を規定し, オブジェクトが持つaspect集合の可能な変化を妥当な範囲に制限する。これに対して, OM/BCサブモデルはオブジェクトが持つaspect集合を妥当な範囲に制限する。

OM/BSサブモデルとOM/BCサブモデルの構築には, 高水準ベトリネットである述語/遷移ネット(Predicate/Transition Net) [14]を拡張したクラス/遷移ネット(Class/Transition Net)が使われる。OM/BSサブモデルに対するクラス/遷移ネット(OM/BSネット)では, プレースノードがクラススキーマに, プレースノード内のトークンが当該クラススキーマが定義するaspectインスタンスに, 遷移ノードがオブジェクトに対するaspectの追加・削除を伴うOM事象に, トークン集合がオブジェクトが持つaspect集合の状態に, 各々対応付けられる。一方, OM/BCサブモデルに対するクラス/遷移ネット(OM/BCネット)では, 新たに制約違反状態を表わすプレースが加わり, 遷移ノードは制約違反検出のための検証事象に対応付けられる。

クラス/遷移ネットはOMの振舞いモデルに対する強力な記述力, 図式表現によるモデルの理解の容易さ, シミュレーションによる実行可能モデルといった性質を有する。OODBの振舞いモデルに対する従来研究でもプレース/遷移ネットといった(低水準)ベトリネットが応用されてきた[15]。しかし, クラス/遷移ネットとは異なり, 低水準ベトリネットはデータ構造を扱うことができず, オブジェクトの属性値に関する条件に基づく振舞い表現には不十分である。一方, 情報システムのモデル化研究において, 高水準ベトリネットの応用により複雑なデータ構造を扱った事例もある。文献[16]では述語/遷移ネットにより関係スキーマが, 文献[17]では非正規関係/遷移ネット(Nested-Relation/Transition Net)により非正規関係スキーマが扱われている。しかし, 前者では属性が単純値に制限され, 後者では属性がオブジェクト識別子をとることはなく, いずれのモデルもクラス/遷移ネットが対象とするオブジェクト構造を扱うことはできない。クラス/遷移ネットはデータ構造としてクラススキーマを扱うことができ, かつ, このデータ構造を踏まえた遷移の表現を可能とする。

文献[18]では, 規則集合によるOM/BSサブモデルの表現の枠組みが提案されている。規則集合によるモデル記述は, クラス/遷移ネットのような図式表現に比較してモデルの全体的な把握が困難である。また, この枠組みでは, 規則が条件付き・文脈依存の範囲の制約を扱うことができないし, OM/BCサブモデルによる扱う制約を扱うこともできない。文献[19]は著者等が提案したOMに対する枠組みであり, 本論文で扱うOM振舞いモデルの範囲の制約を表現することができる。しかし, [18]と同様に, その表現には規則形式が用いられており, 本論文ではモデルの理解の容易さといった点で拡張を行っている。

本論文の構成は, 以下の通りである。2. ではMAORIの概略的な説明を, 3. ではOM振舞いモデルに対する要件を述べる。この下で, 4. ではOM振舞い構造ネットを, 5. ではOM振舞いチェックネットを提案する。6. では, OM振舞いモデルの例を示す。最後に, 7. で本論文のまとめと今後の課題を示す。

2. 多面的オブジェクト指向データモデル

多面的オブジェクト指向データモデルMAORI [12]では, 実世界の实体は複数aspectを持つオブ

ジェクトとしてモデル化される。オブジェクトの各aspectはクラスのインスタンスとして表現され、クラスはインスタンスの形式(型)として属性/メソッドのリストを定義する。クラス間には上位(下位)クラス関係が存在し、下位クラスは上位クラスの定義を継承する。オブジェクトは、それが持つaspectを定義する全てのクラスに所属する。オブジェクトのクラスへの所属関係は、オブジェクトに対するaspectの追加・削除に伴って動的に変更される。オブジェクトは、オブジェクト識別子(object identifier; oid)により一意に識別される。また、aspectはaspect識別子(aspect identifier; aid)によりオブジェクトにおいて一意に識別される。

図1は、オブジェクトの例を示す。このオブジェクトは人を表わしており、@STUDENT, @WORKER, @MANAGERといったクラスが定義するaspectを有している。特筆すべき点は、図1のオブジェクトがaspect @MANAGERを2つ持っている点にある。このように、同一のクラスが定義するaspectを複数持つことができる。これは、[1]-[5]にはないMAORIの特徴である。

3. OM振舞いモデルに対する要件

多面的オブジェクト指向データベースにおけるOM振舞いモデル構築のためには種々の要件の考慮が必要となる。以下に、これらの要件を例とともに示す。

[要件1] OMは、オブジェクトのaspect集合を変化させる事象(event)、事象に関連するaspectを指定する状態(state)、さらにこれらの事象と状態

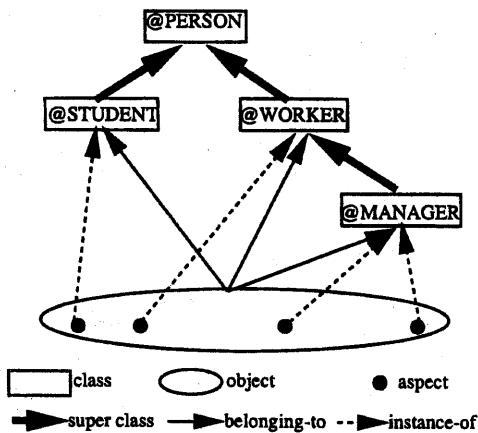


図1 多面的オブジェクト
Fig.1 A multi-aspects object.

との組合せによって定義される。例えば、大学進学といった事象は、ある人間の高校生aspectを削除し、大学生aspectを追加することになる。これら要素ならびに要素間の関係が、モデル上で表現されなければならない。

[要件2] オブジェクトが持つことができるaspectの型は制限されている。犬や猫はペットとしてのaspectを持つことができる。しかし、人間は比喩的な意味を除けばペットとしてのaspectを持つことはできない。

[要件3] オブジェクトのaspect間の遷移には方向の制限が課せられる。高校生から大学生になることはできるが、大学生から高校生になることはできない。

[要件4] オブジェクトのaspect間の遷移には、条件が課せられる。大学生は所定の単位を取得した後に、卒業生になることができる。

[要件5] オブジェクトのaspect間の遷移は、文脈に依存する。助教授は研究者としてのaspectを持ち、かつ所定数以上の論文を書いていることを条件に教授になることができる。この場合、助教授aspectから教授aspectへの遷移は、ある条件を満たす研究者aspectの存在といった文脈に依存している。

[要件6] Aspectの型毎に、オブジェクトが持つことができるaspect数には上限がある。例えば、ある会社では課長は2つまで兼任することができる。

[要件7] オブジェクトには、同時には共存できないaspectの型の組が存在する。ある人間が同時にある大学の在大学生であり、かつ卒業生であることはない。

[要件8] オブジェクトのaspectの属性に関して、常に満たすべき条件が課せられる。例えば、ある大学では本人の収入が一定額を超えると、学生は奨学金を支給されない。

[要件1]は、OM振舞いモデルに対する基本的要件である。[要件2]から[要件8]までは、オブジェクトのaspect集合に対して、OMに際して維持されるべき一貫性制約に関する。これらの一貫性制約は、(1)静的制約、すなわちオブジェクトの可能なaspect集合を制限する制約、(2)遷移制約、すなわちオブジェクトが持つaspect集合の可能な変化を制限する制約に分類される。上記の[要件2]、[要件6]、[要件7]、[要件8]は静的制約の例であり、[要件3]、[要件4]、[要件5]は遷移制約の例である。

一方、OM振舞いモデルに対する一貫性制約の取り込みには、2つのアプローチがある。第1

のアプローチでは、一貫性制約に従うOMだけが構造的に生起するOM振舞い構造ネットを構築する。第2のアプローチでは、OMに際して一貫性制約に違反する状況が生じた時にこれを検出するOM振舞いチェックネットを構築する。本論文では、OM振舞いモデルの構築のために、これら2つのアプローチを併用する立場をとる。すなわち、[要件1]、[要件2]ならびに特定のOMのタイプに帰着可能な一貫性制約に関する[要件3]、[要件4]、[要件5]に対しては、OM振舞い構造ネットを構築する。それ以外の[要件6]、[要件7]、[要件8]の静的制約は、複数のOMに関係することになる。従って、OM振舞い構造ネットへのこれらの静的制約の取り込みは、個々のOM記述に複雑さと重複記述をもたらすため問題である。このため、本論文では[要件6]、[要件7]、[要件8]に対するOM振舞いモデルとして、OM振舞いチェックネットを構築することとする。

4. OM振舞い構造ネット

ペトリネット(Petri Net) [13]は、C.A.Petriにより提案された並行的・分散的なシステムを表現するための数学的モデルである。ペトリネットは、互いに素な集合であるプレースノード集合と遷移ノード集合とからなる2部有向グラフ(bipartite directed graph)である。一般に、プレースノードは状態に関連したシステム要素を、遷移ノードはアクティビティに関連したシステム要素をモデル化する。ペトリネットは、システムの振舞い構造を非形式的に記述する。しかし、プレースノード/遷移ノードといった単純なペトリネットの基本モデルは、システムの状態あるいはシステムの振舞いに関する形式的概念を備えていない。このため、基本的なペトリネット概念を拡張した表現力にとんだ高水準ペトリネットが存在している。

述語/遷移ネット[14]では、例えばプレースノードが関係スキーマ(述語)を表わし、プレースノード/遷移ネットに比して複雑なデータ構造の扱いが可能となっている。しかし、関係スキーマの属性は単純値に制限されており、オブジェクト構造を扱うことはできない。このため、OM振舞いモデルの構築に向けて、述語/遷移ネットを拡張したクラス/遷移ネットを提案する。クラス/遷移ネットではプレースノードはクラススキーマを表わし、クラス名が付される。クラス/遷移ネットはオブジェクト単位でインスタンス化され、各プレースノードには当該クラススキーマが定義するaspectインスタンス、いわゆるマーキングが関数により割り当てられる。ある時点における全ての

プレースマーキングの集合は、そのクラス/遷移ネットに対応するオブジェクトのaspect集合の状態を表わす。遷移ノードは入力/出力プレースのaspectが関係するOM事象を表わす。遷移ノードが発火すると、その入力プレースからaspectが削除され、その出力プレースにaspectが挿入される。遷移ノードには、入力プレースのaspectに関する発火条件を、出力プレースのaspectに関する生成仕様を与えることができる。また、プレースノードと遷移ノードとの間のアークにはラベルが付され、これらは発火条件、生成仕様の記述においてaspectを参照するために使われる。尚、アークに付されるラベル名は、そのアークが入る(または出る)遷移ノードを有効範囲(scope)とする。また、遷移ノードの発火には、外部からパラメータを与えることもできる。このパラメータの値は、同様に発火条件あるいは生成仕様で参照可能である。

図2(a)のOM振舞い構造ネットは、助教授から教授への昇進OMをモデル化する。クラス/遷移ネットでは、プレースノードは丸(○)で、遷移ノードは四角形(□)または棒(|)で表わされる。遷移ノードに入るアークの始点にあるのが入力プレースであり、遷移ノードから出るアークの終点にあるのが出力プレースである。遷移ノードの四角形の内部には、発火条件(condition句)と生成仕様(output句)が与えられる。図2(a)のOM遷移(OM transition)ノードの発火条件は、@RESEARCHER aspectを参照するためアークに付けられたラベルRを使っている。この条件は、助教授から教授への昇進には、研究者として所定数以上の論文を書いていることを必要としている。特筆すべきことに、発火条件には入力プレース

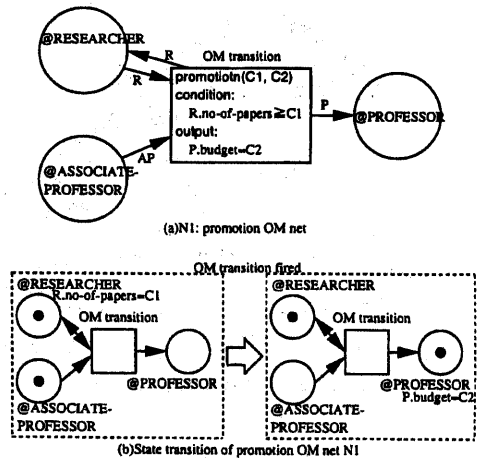


図2 OM振舞い構造ネット
Fig.2 OM behavior structure network.

スの@RESEARCHER aspectと@ASSOCIATE-PROFESSOR aspectとを関係付ける結合述語(join predicate)がない。これは、クラス/遷移ネットがオブジェクト単位でインスタンス化されるため、任意のオブジェクトの@RESEARCHER aspectと@ASSOCIATE-PROFESSOR aspectとが暗黙的にグループ化されており、結合述語が不要となっているからである。尚、この例ではパラメータC1, C2が各々、発火条件、生成仕様で参照されている。

遷移ノードで与えられる生成仕様は、出力プレースに挿入される各aspectの属性の値を規定する。このため、出力プレースのaspectの属性値と入力プレースのaspectの属性値との関係が式により与えられる。図2(a)では、新たに追加される@RESEARCHER aspectの属性budgetの値が与えられている。一方、生成仕様において未指定の属性値は、次の規則によって規定される。

- ・ 入力プレースのaspectの属性と同一の祖先となるクラスで定義される属性には、入力プレースのaspectの当該属性値が伝播される。

- ・ それ以外の属性には、NIL値が与えられる。

さらに、生成仕様において未指定の出力プレースのaspectに対しては、次のように暗黙的な生成仕様が適用される。

- ・ プレースノードが遷移ノードの入出力プレースであり、かつプレースノードから遷移ノードへのアークと遷移ノードからプレースノードへのアークに同じラベルが付られている場合には、入力プレースのaspectがそのまま生成される。図2(a)の@RESEARCHER aspectは、この例である。こうしたaspectは、OM生起の文脈としての役割を担っている。

- ・ それ以外のaspectの場合、入力プレースのaspectの属性と同一の祖先となるクラスで定義された属性には、入力プレースのaspectの当該属性値が伝播される。それ以外の属性には、NIL値が与えられる。

図2(b)は、図2(a)のOM振舞い構造ネットの状態遷移例を示す。プレースノード内の黒丸(●)は、aspectインスタンスを表わすトークンである。@RESEARCHER aspectと@ASSOCIATE-PROFESSOR aspectとが生起条件を満たし、OM遷移ノードが発火すると、@ASSOCIATE-PROFESSOR aspectのトークンが消え、@PROFESSOR aspectのトークンが生成される。このとき、この@PROFESSOR aspectの属

性budgetの値はC2となる。

5. OM振舞いチェックネット

OM振舞い構造サブモデルの構築と同様に、OM振舞いチェックサブモデルの構築にはクラス/遷移ネットを応用する。このためのクラス/遷移ネットは、OM振舞いチェックネットとよばれる。OM振舞いチェックネットでは、静的制約に対する検査を制御する制御遷移(control transition)ノードと制御プレース(control place)、静的制約を検査するチェック遷移(check transition)ノード、静的制約違反の有無を示す制約違反プレース(constraint violation place)と制約違反プレースのトークンを削除する清掃遷移(cleaning transition)ノードが導入される。制御プレース、制約違反プレースはプレース/遷移ネットのプレースノードに同じであり、マーキングされるのはデータ構造を持たない単なるトークンである。

図3(a)のOM振舞いチェックネットは、aspectの上限数に関する制約(3.の[要件6])をモデル化する。図3(a)では、チェック遷移ノードは生起条件、生成仕様を持っていない。この遷移ノードは、制御プレースにトークンがマーキングされ、かつ課長を3つ以上兼任するような状態において発火可能となる。チェック遷移ノードが発火すると制約違反プレースにトークンがマーキングされる。@MANAGERプレースとチェック遷移ノード間の入力アーク集合と出力アーク集合には、M1, M2, M3とラベル付けされた対となるアークが含まれる。すなわち、制約違反状態の検出のために入力プレース(@MANAGER)から削除されたaspectは、元通りに出力プレース(@MANAGER)に戻される。また、制御プレースは、チェック遷移ノードが何度も発火して、制約違反プレースにトークンが無限に増加することを防止する。図3(b)は、図3(a)のOM振舞いチェックネットの状態遷移例を

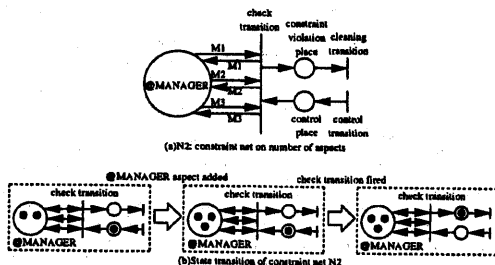


図3 OM振舞いチェックネット
Fig.3 OM behavior check model.

示す。@MANAGER aspectのトークンが2つ存在している状態において、3つ目のトークンが挿入されると、チェック遷移ノードが発火して、制約違反プレースにトークンが挿入される。

図4(a)のOM振舞いチェックネットは、排他的aspectに関する制約(3.の[要件7])をモデル化する。図3(a)と同様に図4(a)では、チェック遷移ノードは発火条件、生成仕様を持っていない。この遷移ノードは、制御プレースにトークンがマーキングされ、かつ@STUDENT aspectと@GRADUATE aspectとが同時に存在するような状態において発火可能となる。チェック遷移ノードが発火すると制約違反プレースにトークンがマーキングされる。尚、図4(a)の例は、在学生と卒業生といった2つのクラス間の排他関係を扱っているが、これを3つ以上のクラス間の排他関係へ一般化することは自明である。図4(b)は、図4(a)のOM振舞いチェックネットの状態遷移例を示す。@STUDENT aspectのトークンが存在している状態において、@GRADUATE aspectのトークンが挿入されると、チェック遷移ノードが発火して、制約違反プレースにトークンが挿入される。

図5(a)のOM振舞いチェックネットは、aspectの属性値条件に関する制約(3.の[要件8])をモデル化する。図5(a)のチェック遷移ノードの発火条件は、「学生本人の収入が一定額(この値はパラメータC3によって外部から与えられる)を超えており、かつ奨学金を支給されている」といった状態を示す。チェック遷移ノードは、制御プレースにトークンがマーキングされ、かつ発火条件を満足する@STUDENT aspectと@WORKER aspectとが存在するような状態において発火可能となる。チェック遷移ノードが発火すると制約違反プレースにトークンがマー

キングされる。また、チェック遷移ノードの生成仕様は、制約違反を解消するために、「この学生の奨学金の支給を停止する」ように属性scholarshipの値を変更している。図5(b)は、図5(a)のOM振舞いチェックネットの状態遷移例を示す。@WORKER aspectのトークンが存在している状態において、@STUDENT aspectのトークンが挿入され、かつ生起条件が満たされると、チェック遷移ノードが発火して、制約違反プレースにトークンが挿入される。このとき、@STUDENT aspectの属性scholarshipの値は変更される。

6. OM振舞いモデルの例

図1に示す多面的オブジェクトに関して、図6にOM振舞いモデルの例を示す。これ迄、OM振舞いモデルに関して、個々の遷移制約をモデル化するOM振舞い構造ネット、個々の静的制約をモデル化するOM振舞いチェックネットを個別に述べてきた。しかし、図6(a)に示すように、オブジェクトが保持しうるaspectのクラス集合が関係する範囲でのOM振舞い構造ネット、OM振舞いチェックネットを併合したものが、本来的なOM振舞いモデルとなる。すなわち、OM振舞いモデルはクラスに対応するプレースノード集合をベースとして、その下に遷移制約、静的制約に対応するネットワーク構造を作りあげたものとなる。図6(a)のマーキングの状態より、このオブジェクトは課長を2つ兼任している会社員を表わしており、かつこの人は社会人学生であることになる。また、このOM振舞いモデルには、次の遷移制約、静的制約が含まれている。

- [遷移制約] OM1: 学生の卒業に関する。
- OM2: 課長への昇進(あるいは兼任)に関する。

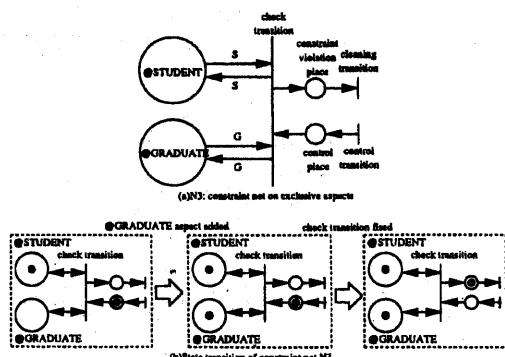


図4 OM振舞いチェックネット
Fig.4 OM behavior check network.

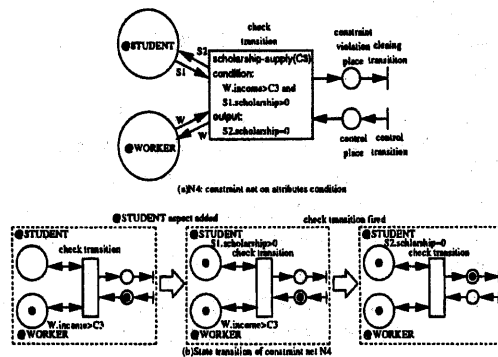


図5 OM振舞いチェックネット
Fig.5 OM behavior check network.

OM3: 社員になることに関する。

- [静的制約] ct1~ct3: @GRADUATE, @STUDENT, @WORKERの各 aspectは高々1 aspectずつしか保持できない。
 ct4: @MANAGER aspectは高々2 aspectしか保持できない。
 ct5: @STUDENT aspectと @GRADUATE aspectとは排他関係にある。

さて、図7(a)の状態において、この社員が3つ目の課長を兼任し、それに伴い収入(給与)が増加し、その結果、奨学金支給の基準額を超えたものとする。図7(b)は、その設定の下での、図7(a)からの状態変化部分だけを示している。すなわち、3つ目の課長を兼任することによる制約違反(cvp4ノード)と奨学金支給に

関する制約違反(cvp6ノード)が同時に検出されている。

図7よりクラス/遷移ネットが以下の性質を有することがわかる。

- 第3章で述べたOM振舞いモデルの要件を満たす強力な記述力を持っている。
- 図式表現によるモデルは、制約記述のための規則集合に比較して、制約表現を構成する個々の要素とそれらの間の関係を全体的に把握することを可能とする。これは、モデルに対する理解を容易とする。
- 与えられたマーキングの下で、モデルを使った状態変化のシミュレーションをすることが可能である。これはクラス/遷移ネットが実行可能モデルであることを意味し、そのためのシミュレータの実現を要請するものである。

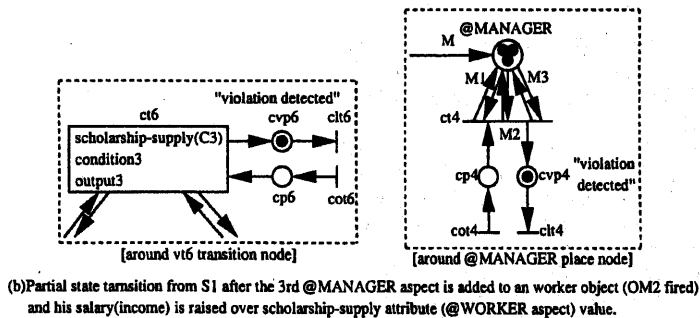
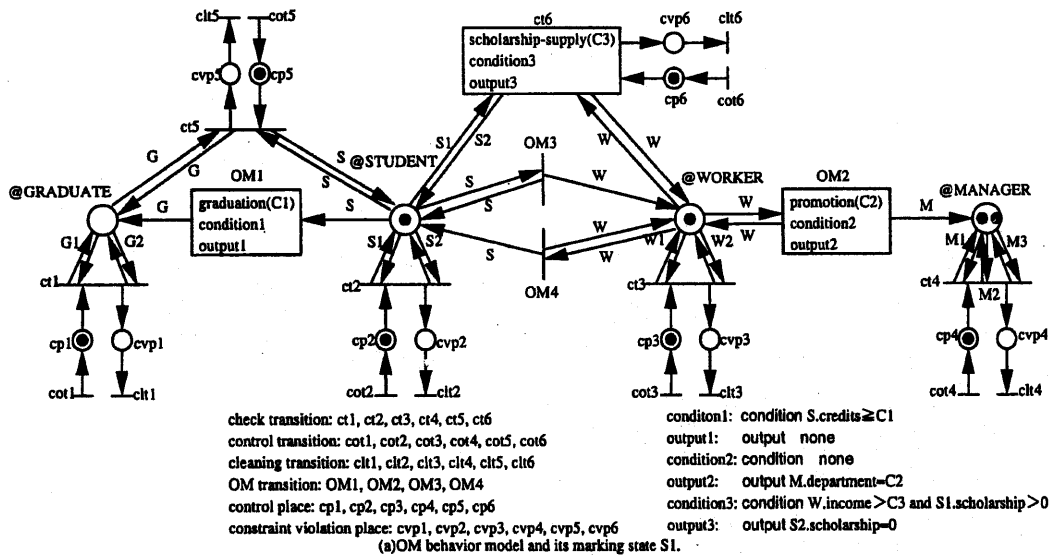


図6 OM振舞いモデル例
 Fig.6 An example of OM behavior model.

7. おわりに

本論文では、多面的オブジェクト指向データモデルMAORIに基づくデータベースにおいて、ペトリネット応用によるOM振舞いモデルの構築を提案した。モデル化のために述語/遷移ネットを拡張したクラス/遷移ネットを提案し、これに基づき、遷移制約に対するOM振舞い構造ネット、静的制約に対するOM振舞いチェックネットを構築した。本モデル化の枠組みは、以下の特徴を持っている。

- ・ クラス/遷移ネットは、オブジェクト構造を扱うことが可能である。

- ・ クラス/遷移ネットを応用したOM振舞い構造ネットとOM振舞いチェックネットは、OM振舞いモデルに対する十分な記述力、図式表現によるモデルの理解の容易さ、シミュレーションによる実行可能モデルといった性質を有する。

今後の課題としては、(1)OM振舞いモデルエディタ、シミュレータ等の試作、(2)OM振舞いモデルの冗長性、無矛盾性チェックの検討、(3)制約違反生起に対する調整機構の検討などを考えている。

文 献

- [1] E. Sciore : "Object Specialization", ACM Trans. Off. Inf. Syst., vol.7, no.2, pp.103-127, 1989.
- [2] L. A. Stein and S. B. Zdonik : "Clovers: The Dynamic Behavior of Type and Instances", Brown University Technical Report No.CS-89-42, 1989.
- [3] J. Richardson and P. Schwarz : "Aspects: Extending Object to Support Multiple, Independent Roles", Proc. of the 1991 ACM SIGMOD Int. Conf. Manage. Data, pp.298-307, 1991.
- [4] 塚田晴史, 杉村利明 : "MACモデル: 複数観点からの分類が可能なオブジェクト", コンピュータソフトウェア, vol.11, no.5, pp.44-57, Sept. 1994.
- [5] 石丸知之, 植村俊亮 : "オブジェクト指向データモデルにおけるオブジェクトの多重表現", 信学論(D-I), vol.J78-D-I, no.3, pp.349-357, Mar. 1995.
- [6] M. L. Brodie and D. Ridjanovic : "On the Design and Specification of Database Transactions", On Conceptual Modelling, eds. M. L. Brodie, J. Mylopoulos and J. W. Schmidt, pp.277-306, Spring-Verlag, 1984.
- [7] J. Mylopoulos, P. A. Bernstein and H. K. T. Wong : "A Language Facility for Designing Database-Intensive Applications", ACM Trans. Database Syst., vol.5, no.2, pp.185-207, Jun. 1980.
- [8] F. Put : "The ER Approach Extended with the Actiton Concept as a Conceptual Modeling Tool", Proc. of the 7th Int. Conf. Entity-Relationship Approach, 1988.
- [9] H. Sakai : "A Method for Entity-Relationship Behavior Modeling", Proc. of the 3rd Int. Conf. Entity-Relationship Approach, 1983.
- [10] H. Sakai and H. Horiuchi : "A Method for Behavior Modeling in Data Oriented Approach to System Design", Proc. of the 1st Int. Conf. Data Engineering, 1984.
- [11] M. Schrefl : "Behavior Modeling by Stepwise Refining Behavior Diagrams", Proc. of the 9th Int. Conf. Entity-Relationship Approach, 1990.
- [12] 佐藤秀樹, 池田峰輝, 舟橋栄, 林達也 : "多面的オブジェクト指向データモデルMAORI", 信学論(D-I), vol.J79-D-I, no.10, pp.781-790, Oct. 1996.
- [13] J. L. Peterson : "Petri Net Theory and the Modeling of Systems", North-Holland, 1981
- [14] K. Jensen and G. Rozenberg : "High Level Petri Nets", Springer-Verlag, 1991.
- [15] 酒井博敬 : "オブジェクト指向データベース設計", 情報処理, vol.32, no.5, pp.568-576, May 1991.
- [16] H. J. Genrich and K. Kautenbach : "System Modeling with high level Petri Ntes", Theor. Comput. Sci., 13, pp.109-136, 1981.
- [17] A. Oberweis and P. Sander : "Information System Behavior Specification by High-Level Petri Nets", ACM Trans. Inf. Syst., vol.14, no.4, pp.380-420, Oct. 1996.
- [18] L. Qing and D. Guozhu : "A framework for object migration in object-oriented databases", Data & Knowledge Eng., 13, pp.221-242, 1994.
- [19] 佐藤秀樹, 舟橋栄, 林達也 : "多面的オブジェクトに対するオブジェクトマイグレーションの枠組み", 信学論(D-I), (採録済・掲載予定)