

多面的オブジェクト指向データベースにおける オブジェクトマイグレーションの枠組み

佐藤秀樹* 舟橋栄** 林達也**

*日本電装株式会社

**名古屋工業大学電気情報工学科

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けるオブジェクトマイグレーション(Object Migration)の枠組みを提案する．実世界において意味のない，あるいは誤ったオブジェクトマイグレーションの生起を回避するための制御仕様としてオブジェクトスキーマと遷移規則を用いる．オブジェクトスキーマは静的制約の定義であり，オブジェクトの基本的な性質を表わす核aspectを通してオブジェクトに動的に関連付けられる．また，遷移規則は動的制約を定義する書換え規則であり，条件付き，文脈依存といった点を特徴としている．

On the Framework of Object Migration in Multi-Aspects Object-Oriented Databases

Hideki Sato * Sakae Funahashi ** Tatsuya Hayashi **

*NIPPONDENSO CO., LTD.

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

This paper proposes a framework of object migration in databases based on MAORI(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. The framework uses object schemas and migration rules as control specifications to avoid meaningless and/or invalid migration in the real-world. Object schemas define static constraints and are dynamically related to objects by core-aspects representing objects' fundamental properties. Furthermore, migration rules, a kind of rewriting rules, are used to define dynamic constraints. The form of rules is descriptive in both conditional and context-sensitive features.

1. はじめに

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-Aspects Object-Oriented Data Model)[1]に基づくデータベースにおけるオブジェクトマイグレーション (Object Migration; 以降ではOMと記す)の枠組みを提案する。OMは、オブジェクトのクラスへの所属関係の更新を意味する。提案する枠組みは、実世界において意味のない、あるいは誤ったOM生起を回避するための制御仕様としてオブジェクトスキーマと遷移規則を用いる。オブジェクトスキーマは静的制約の定義であり、オブジェクトの基本的な性質を表わす核aspectを通してオブジェクトに動的に関連付けられる。また、遷移規則は動的制約を定義する書換え規則であり、文献[2]の制御規則の形式より(1)条件付き、(2)文脈依存といった点で記述能力が高くなっている。

本論文の構成は、以下の通りである。2. では、例を通して実体に関する多面性制約を考察する。3. ではMAORIを、4. ではMAORIに基づくデータベースにおけるOMモデルを示す。その下で、多面性制約の扱いを考慮したOM実現のため、5. で制御仕様の形式化を、6. で制御機構の形式化を行う。最後に、7. で本論文のまとめと今後の課題を述べる。

2. 多面的制約

実世界の実体をモデル化するオブジェクトは、実体の意味を反映した種々の一貫性制約に従うことが要請される。これらの一貫性制約の中で実体の多面性に関連するものを多面性制約とよぶ。以下では、例を通して多面性制約を示す。

[制約例 1] 犬や猫といった実体はペットとしてのaspectを持つことができる。しかし、人間といった実体は、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。すなわち、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。

[制約例 2] ある会社では、課長は2つまで兼任することができる。このように、実体は同じ型のaspectを複数持つことがあるが、その場合にも型毎にオブジェクトが持つことができるaspectの上限数には制約が課せられる。

[制約例 3] 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。すなわち、オブジェクトには同時に共存できないaspectの型の組が存在する。

[制約例 4] 大相撲の力士が関脇から大関になったり、逆に大関から関脇になることはできる。しかし、大関から横綱になることはできるが、横綱から大関になることはできない。このように、オブジェクトのaspect間の遷移には可逆的な場合と非可逆的な場合がある。

[制約例 5] 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。すなわち、aspect間の遷移には条件が課せられる。

[制約例 6] 助教授は研究者としてのaspectを持ち、かつ一定以上の研究成果をあげ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ことになる。

[制約例 1], [制約例 2], [制約例 3]は、オブジェクトの生存期間における任意の時点においてオブジェクトの多面的な性質を表現するaspect集合が満たすべき制約に関する。これらを静的制約とよぶ。これに対して、[制約例 4], [制約例 5], [制約例 6]はオブジェクトのaspectの妥当な獲得・喪失に関する制約に関する。これらを動的制約とよぶ。

3. 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[1]の基本要素は複数クラスに属するオブジェクト、オブジェクトの1つの側面を表わすaspect、aspectの表現であるインスタンス(instance)、インスタンスの枠組の定義とオブジェクトの分類を行うクラス、関連するクラスを集めたクラスグループである。ここでは、以降の議論に必要な範囲でMAORIの定義を与える。

[定義3.1] インスタンスの形式は、 $\{a_1=v_1, \dots, a_n=v_n\}$ である。ここで、 $a_i, v_i (1 \leq i \leq n)$ は、それぞれ属性名と属性値である。□

属性は、構造を表わすデータ属性と振舞いを表わすメソッド属性に分かれる。メソッド属性の場合の属性値は、メソッド実行の戻り値である。以降では、データ属性とメソッド属性とを区別せず、これらを包括的に属性として扱う。

[定義3.2] クラスはインスタンスの形式 (型) を定義する。□

[定義3.3] クラスは上位クラス関係(super class)を通して、他のクラスの定義を継承することができる。ここで、定義を参照されるクラスを上位クラス、定義を参照するクラスを下位クラスとよぶ。また、直接または間接の上位クラスを祖先とよぶ。□

クラス集合は上位クラス関係の下での半順序集合であり、束構造をなす。

[定義3.4] オブジェクトOは、 $\langle \text{OID}, w \rangle$ の対である。ここで、OIDはオブジェクト識別子であり、オブジェクトの生存期間において不変である。また、wはOのaspect集合 $\{(C_1, I_1), \dots, (C_n, I_n)\}$ であり、AS(O)と表わす。 $C_j (1 \leq j \leq n)$ はクラス名であり、 I_j はクラス C_j が定義するインスタンスである。Oのクラス名の集合 $\{C_1, \dots, C_n\}$ をCS(O)、インスタンスの集合 $\{I_1, \dots, I_n\}$ をIS(O)と表わす。但し、CS(O)は重複 (バグ) 集合であり、CS(O)では同一のクラス名の異なる生起は添字を付して区別する。□

[定義3.5] Oがオブジェクトであり、 $A_j = (C_j, I_j)$, $A_j \in \text{AS}(O)$ であるとき、 $\text{CLASS}(A_j)$ は C_j を、 $\text{INS}(A_j)$ は I_j を表わす。□

[定義3.6] Oはオブジェクトであり、 $A_1, A_2 \in \text{AS}(O)$ であり、かつ $\text{CLASS}(A_1)$ と $\text{CLASS}(A_2)$ とが同一の祖先であるクラスで定義された属性を持つ場合、 $\text{INS}(A_1)$ と $\text{INS}(A_2)$ における当該属性は同一値を持つ。□

[定義3.7] オブジェクトOは、 $\text{unique}(\text{CS}(O))$ の要素と

なるクラスに所属する(belonging-to)。但し、 $\text{unique}(S)$ は重複集合 S の要素の冗長性を排除関数である。□

[定義3.7]から、オブジェクト O がクラス C に所属する場合、 O は C が定義するインスタンスをaspectとして持っていることになる。

クラスは、オブジェクトの分類基準を与える。一方、[定義3.3]より上位クラス関係を通したクラス定義の継承機能は、複数クラスのカテゴリをまとめて新たな分類基準を作成する。従って、クラスに所属するオブジェクトは祖先のクラスのカテゴリを満たす。このため、オブジェクトの集合操作を行うときには、クラスに直接所属するものと間接的に所属するものと2つの集合を考える。クラスは前者の役割を果たすが、後者の役割を持つものとしてクラスグループがある。

[定義3.8] クラスグループ C は、クラス C かその下位クラスの1つに所属するオブジェクトの集合である。□

[定義3.8]から、オブジェクト O がクラスグループ C に所属する場合、 O はクラス C が定義するインスタンス、あるいはクラス C かその下位クラスが定義するインスタンスをaspectとして持っていることになる。

図1は、多面的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aspectを有している。特筆すべき点は、この図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、MAORIのオブジェクトは、同一のクラス(型)が定義するaspectを複数持つことができる。これはMAORIの特徴の一つであり、MAORIを表現力に富むものとしている。

4. オブジェクトマイグレーションモデル

OMは、多面的オブジェクトのクラスへの所属関係の更新を意味する。このため、OMモデルは多面的オブジェクト指向データモデルにおけるオブジェクトに対するaspectの追加・削除に伴う、オブジェクトが保持するaspect集合の変化の過程を対象とすることになる。以下では、MAORIに基づくデータベースにおけるOMモデルを定義する。OMモデルでは時間軸上でのオブジェクトを議論するため、まず時刻の定義から始める。

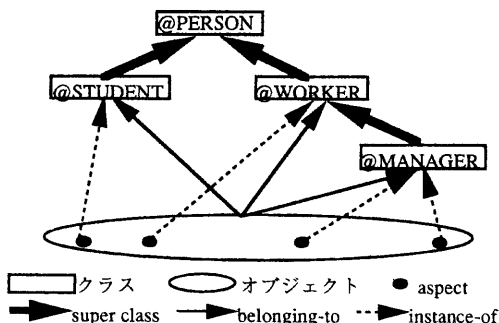


図1 多面的オブジェクト
Fig.1 A multi-aspects object.

[定義4.1] 時刻は、無限個の離散的な値からなる全順序集合の要素である。2項演算子 $=, <, >$ は、時刻の一致や大小関係を比較し真偽値を返す。また、 $\text{preceding}(t), \text{next}(t)$ は、各々、時刻 t の直前と直後の時刻を表わす。□

任意時刻におけるオブジェクトの定義は、次のようになる。

[定義4.2] 時刻 t におけるオブジェクト O を O_t と記す。

但し、 $\text{begin-of}(O) \leq t \leq \text{end-of}(O)$ である。ここで、 $\text{begin-of}(O)$ は O の生成時刻を、 $\text{end-of}(O)$ は O の削除時刻を表わす。□

次に、任意時刻におけるオブジェクトの基本的な性質を表わすaspectとして核aspectを導入する。

[定義4.3] 時刻 t におけるオブジェクト O の核aspectは $\text{AS}(O_t)$ に含まれる特別なaspectであり、 $c\text{-aspect}(O_t)$ と表わされる。任意の時刻において、オブジェクトは唯一の核aspectを持つ。また、核aspect以外のaspectを衛星(satellite) aspectとよぶ。 $c\text{-aspect}(O_{\text{begin-of}(O)})$ は、 O が生成された時点での唯一のaspectを表わす。□

オブジェクトに対するaspectの追加・削除は、オブジェクトの生存期間に生起するOMイベントの副作用による。

[定義4.4] OMイベント e の生起は瞬間的(instantaneous)であり、 $\text{instant}(e)$ は e の生起時刻を表わす。□

[定義4.5] オブジェクト O に対するOMイベント e は、 $\langle \text{AS}(O_{\text{preceding}(\text{instant}(e))}), \text{AS}(O_{\text{instant}(e)}) \rangle$ の2項組として定義される。□

[定義4.5]は、OMイベント e により、オブジェクト O のaspect集合が $\text{AS}(O_{\text{preceding}(\text{instant}(e))})$ から $\text{AS}(O_{\text{instant}(e)})$ に変化することを意味する。

[定義4.6] オブジェクト O に対するOMイベント e により追加されるaspect集合、削除されるaspect集合は次のように定義される。

$$\begin{aligned} \text{add-aspects}(O, e) &= \text{AS}(O_{\text{instant}(e)}) - \text{AS}(O_{\text{preceding}(\text{instant}(e))}) \\ \text{delete-aspects}(O, e) &= \text{AS}(O_{\text{preceding}(\text{instant}(e))}) - \text{AS}(O_{\text{instant}(e)}) \end{aligned} \quad \square$$

表1は、 $\text{add-aspects}(O, e), \text{delete-aspects}(O, e)$ に基づき、OMイベントの性質を分類している。この分類の軸は、[2]の構成的(constructive)、破壊的(destructive)と

表1 マイグレーション・イベントの性質

Table 1 Nature of migration events.

$\text{add-aspects}(O, e)$	$= \phi$	$\neq \phi$
$\text{delete-aspects}(O, e)$	$\text{AS}(O_{\text{preceding}(\text{instant}(e))}) = \text{AS}(O_{\text{instant}(e)})$ であり、副作用なし。	aspectを追加する。
$\neq \phi$	aspectを削除する。	aspectの追加・削除を含み、aspectの型を変更する。

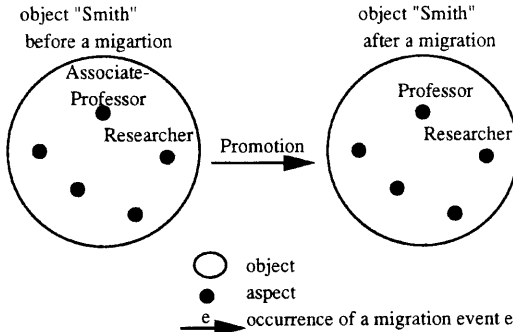


図2 オブジェクトマイグレーション・イベントの例
Fig.2 An example of an object migration event.

いった性質に対応している。図2はOMイベントの例であり、オブジェクトSmithが助教授から教授へ昇進することに関する。

5. オブジェクトマイグレーション制御仕様

実世界の実体の意味を反映したOM実現のためには、多面性制約の扱いを考慮する必要がある。このための負荷を利用者・応用プログラムから軽減するためには、データベース管理システムがOM管理機能を備えることが必要である。OM管理機能は、(1)多面性制約を定義するOM制御仕様、(2)OM制御仕様に基づくOM制御機構からなる。以下では、MAORIに基づくデータベースに対するOM制御仕様としてオブジェクトスキーマと遷移規則を示す。

5.1 オブジェクトスキーマ

オブジェクトスキーマは、(1)オブジェクトが持つことができるaspectの型の制限、(2)オブジェクトが同時には持つことができないaspectの型の組、(3)オブジェクトが持つことができるaspectの型毎のaspectの上限数に関する静的制約を定義する。

[定義5.1] オブジェクトスキーマOSは、 $\{T_1, \dots, T_n\}$ と定義される。ここで、 $T_i = (C_{i(1)}^{r(i(1))} \mid \dots \mid C_{i(k(i))}^{r(i(k(i)))})$ ($1 \leq i \leq n$)であり、 $C_{i(1)}, \dots, C_{i(k(i))}$ はクラス名を、 $r(i(1)), \dots, r(i(k(i)))$ はクラスが定義するaspectの上限数を表す。また、 $C_{i(l)}$ は $C_{i(m)}$ の上位クラスでも下位クラスでもない($1 \leq l, m \leq k(i), l \neq m$)。□

[定義5.1]の T_i ($1 \leq i \leq n$)は、OSに從うオブジェクトが持つことができるaspectを指定する。 T_i に関して各 $C_{i(j)}^{r(i(j))}$ ($1 \leq j \leq k(i)$)は排他的な指定であり、 $C_{i(j)}$ あるいはその下位クラスが定義するaspectを高々 $r(i(j))$ 個持つことができることを表す。

[オブジェクトスキーマ例]

#PERSON = $\{ (@STUDENT^1 \mid @GRADUATE^1),$
 $(@WORKER^1), (@MANAGER^2) \}$

#PERSONに從うオブジェクトは、クラス

@STUDENT, @GRADUATE, @WORKER,

@MANAGERおよび各々の下位クラスのインスタンス

をaspectとして持つことができる。但し、STUDENTと@GRADUATEのインスタンスは、同時にこのオブジェクトのaspectとはならない。また、aspect数に関しては、@MANAGERのインスタンスを2つまで、@STUDENT, @GRADUATE, @WORKERのインスタンスを各々1つだけ持つことができる。

[定義5.2] 時刻tにおけるオブジェクトOがオブジェクトスキーマOSを満たすとき、述語os-sat(O_t, OS)が成立する。□

次に、オブジェクトスキーマ間の上位/下位オブジェクトスキーマ関係を定義する。但し、定義には次の関数fと述語super-class-ofを用いる。

$$f(C_0, (C_1^{r(1)}, \dots, C_k^{r(k)})) = \max(g(C_0, C_1^{r(1)}), \dots,$$

$$g(C_0, C_k^{r(k)}) = \begin{cases} r(q) & (C_p \text{ が } C_q \text{ の上位クラスである場合}) \\ 0 & (\text{それ以外}) \end{cases}$$

$\max(x_1, x_2, \dots)$ は $x_1, x_2, \dots$ の中の最大値を返す。

・super-class-of(C_1, C_2) : C_1 は C_2 の上位クラスである。但し、上位クラス関係は反射的(reflexive)であり、 C_1 と C_2 とが等しい場合にも本述語は成立するので注意されたし。

[定義5.3] オブジェクトスキーマOS₁ = $\{T_{11}, \dots,$

$T_{1n(1)}, T_{1i} = (C_{1i(1)}^{r(1i(1))} \mid \dots \mid C_{1i(k(1i))}^{r(1i(k(1i)))})$ ($1 \leq i \leq n(1)$)がオブジェクトスキーマOS₂ = $\{T_{21}, \dots,$

$T_{2n(2)}, T_{2j} = (C_{2j(1)}^{r(2j(1))} \mid \dots \mid C_{2j(k(2j))}^{r(2j(k(2j)))})$ ($1 \leq j \leq n(2)$)の上位オブジェクトスキーマ（等価的にOS2はOS1の下位オブジェクトスキーマ）であるための必要十分条件は、以下の通りである。

$$(1) \forall j((1 \leq j \leq n(2)) \rightarrow \exists i((1 \leq i \leq n(1)) \wedge \forall m((1 \leq m \leq k(2(j)))) \rightarrow \exists h((1 \leq h \leq k(1(i))) \wedge \text{super-class-of}(C_{1h(h)}, C_{2j(m)} \wedge (r_{1h(h)} \geq r_{2j(m)}))))$$

$$(2) \forall i \forall h(((1 \leq i \leq n(1)) \wedge (1 \leq h \leq k(1(i)))) \rightarrow n(2) (r_{1h(h)} \geq \sum_{j=1}^n f(C_{1h(h)}, T_{2j})))$$

j=1

□

文献[1]では、オブジェクトは生成時にそれが従うべきオブジェクトスキーマに関連付けられる。この関連付けは、オブジェクトの生存期間において変わることはない。しかし、aspectの動的な追加・削除によりオブジェクトの性質が変化していくことを考えると、オブジェクト生成時に将来の生存期間にわたるオブジェクトスキーマの決定は困難な場合があり、かつ柔軟性に欠けると言える。これに対して、本論文では、オブジェクトが従うべきオブジェクトスキーマは動的に変更されることにする。このため、クラスにはそれが定義するaspectを核aspectとして保持するオブジェクトが従うべきオブジェクトスキーマの指定を含むことにす

る。そして、核aspectの移動に伴いオブジェクトが従うべきオブジェクトスキーマも変更されることになる。尚、オブジェクトスキーマの指定は、上位クラス関係を通じた継承の対象として扱われる。

以下では、クラスとオブジェクトスキーマの指定との関係を定義する。

[定義5.4] クラスCのオブジェクトスキーマの指定は、 $o\text{-schema}(C)$ である。□

[定義5.5] クラス C_1 がクラス C_2 の上位クラスであるとき、 $o\text{-schema}(C_1)$ は $o\text{-schema}(C_2)$ の上位オブジェクトスキーマである。□

5. 2 遷移規則

遷移規則は妥当なOMイベントを規定する動的制約の定義であり、次のような書換え規則の形式をとる。

[定義5.6] 遷移規則pは、次の形式をとる。

p: $c(\alpha)$
 $\alpha \text{-----} \rightarrow \beta$

ここで、 α ($\in N^+$; 但しNはクラス名の集合を示す)はOMイベント生起前にオブジェクトが保持する(source-aspect列であり、 $\text{left-hand}(p)$ と表わされる。 β

($\in N^*$)はOMイベント生起後にオブジェクトが保持する(target-aspect列であり、 $\text{right-hand}(p)$ と表わされる。 $c(\alpha)$ は、 α のaspectの属性値に関する述語を使った論理的結合条件である。これはOMイベント生起条件とよばれ、 $\text{cond}(p)$ と表わされる。 α 、 β に現われる同一のクラス名は同一のaspectを表わす。但し、同じクラスが定義する、複数の異なるaspectはクラス名に添字を付けて区別される。□

以下に遷移規則の例を示す。

[規則1] 所定の単位数を取得した後に、学生は卒業生になることができる。([制約例5])。

p₁: $\text{@STUDENT.credits} \geq c_1$
 $\text{@STUDENT} \text{-----} \rightarrow \text{@GRADUATE}$
 p₁の形式は、形式言語理論における文脈自由規則

(context-free rule)[3]に相当する。ここで、単位数(属性credits)に関する条件は、文解析における意味チェック仕様に相当する。

[規則2] 助教授は教授になることができる。但し、研究者として研究成果をあげていることが必要である。([制約例6])。

p₂: cond_2
 $\text{@RESEARCHER @ASSOCIATE-PROF} \text{-----} \rightarrow \text{@RESEARCHER @PROFESSOR}$
 $\text{cond}_2: \text{@RESEARCHER.no-of-papers} \geq c_2$

p₂の形式は、形式言語理論における文脈依存規則(context-sensitive rule)[3]に相当する。ここで、 @RESEARCHER は @ASSOCIATE-PROF から @PROFESSOR へのOMのための文脈を規定している。しかし、形式言語理論の文脈依存規則とは異なり、文脈依存の遷移規則pは $|\text{left-hand}(p)| > |\text{right-hand}(p)|$ となるε規則[3]を含むことができる。[規則3] ある会社では、課員数が10人以上の課を担当していなければ、課長を兼任することができる。

p₃: cond_3
 $\text{@MANAGER}_1 \text{-----} \rightarrow \text{@MANAGER}_1 \text{ @MANAGER}_2$
 $\text{cond}_3: \text{@MANAGER}_1.\text{no-of-members} < 10$

p₃では、 @MANAGER_1 、 @MANAGER_2 は、クラス @MANAGER が定義する異なるaspectを表わす。

遷移規則のsource-aspect列、target-aspect列のようなクラス名列に対して、クラス名を要素とする集合を求める関数は次の通りである。

[定義5.7] クラス名列vを構成するクラス名の重複集合は、 $\text{constituent}(v)$ と表わされる。□

[定義5.8] 時刻tにおけるオブジェクトOに対して次の(1)、(2)を満たす関数σが存在するとき、オブジェクトOは遷移規則pの前提条件を満たすといひ、述語 $\text{pre-cond-sat}(O, p)$ が成立する。

(1) $\text{left-cond}(p) = V_1 \dots V_k$, $\text{AS}(O) = \{(C_1, I_1), \dots, (C_n, I_n)\}$ であるとき、σは $\{1, \dots, k\}$ を定義域、 $\{1, \dots, n\}$ を値域とする単射関数である。但し、 $V_j (1 \leq j \leq k)$ と $C_{\sigma(j)}$ は、同一のクラス名を表わす。

(2) 代入 $\Theta = \{I_{\sigma(1)}/V_1, \dots, I_{\sigma(k)}/V_k\}$ を適用した結果、 $\text{cond}(p)$ の条件が成立する。但し、 $I_{\sigma(j)}/V_j (1 \leq j \leq k)$ はクラス名 V_j の生起を $I_{\sigma(j)}$ で置換することを意味する。□

[定義5.8]の(1)は、source-aspect列がOM対象のオブジェクトにおいて存在するための条件である。また、(2)はsource-aspect列に関するイベント生起条件の成立に関する。

6. オブジェクトマイグレーション制御機構

OM管理のための制御機構は、次に定義するスキーマセットの下で多面性制約をチェックする。

[定義6.1] スキーマセットSSは、 $\langle \text{CL}, \text{super-class-of}, \text{OL}, P \rangle$ の4項組によって定義される。ここで、CLはクラスの有限集合、super-class-ofはCL上の上位クラス(半順序)関係、OLはオブジェクトスキーマの有限集合、Pは遷移規則の有限集合である。□

与えられたスキーマセットに関して、オブジェクトOtのaspect集合からOMにより遷移可能なaspect集合を要素とするaspect集合形式の集合は次のように定義される。

[定義6.2] スキーマセットSS($= \langle \text{CL}, \text{super-class-of}, \text{OL}, P \rangle$)の下で、オブジェクト $O_t (= \langle \text{OID}, w_t \rangle)$ のaspect集合形式の集合 $\text{ASF}(SS, O_t)$ は以下のように定義される。

(1) $w_t \in \text{ASF}(SS, O_t)$ である。

(2) もし $w_1 \in \text{ASF}(SS, O_t)$ であり、かつ遷移規則 $p \in P$ があり、次の(2.1)-(2.5)が成立するとき $w_2 \in \text{ASF}(SS, O_t)$ である。

(2.1) $\text{pre-cond-sat}(\langle \text{OID}, w_1 \rangle, p)$ が成立する。

(2.2) $\text{CS}(\langle \text{OID}, w_2 \rangle) = (\text{CS}(\langle \text{OID}, w_1 \rangle) - \text{constituent}(\text{left-hand}(p))) \cup \text{constituent}(\text{right-hand}(p))$ である。

(2.3) $\text{CC}_1 = \text{CLASS}(c\text{-aspect}(\langle \text{OID}, w_1 \rangle))$, $\text{CC}_2 =$

CLASS(c-aspect(<OID, w_2 >))であるとき次が成立する。
但し、述語LUB(C, CS)はクラスCが上位クラス関係の下での束構造におけるクラス集合CSの最小上界LUB (Least Upper Bound)であるとき成立する。

$((CC_1 \in \text{constituent}(\text{left-hand}(p)) \wedge$
 $CC_1 \notin \text{constituent}(\text{right-hand}(p))) \rightarrow$
 $(CC_2 \in \text{constituent}(\text{right-hand}(p)) \wedge$
 $\text{LUB}(CC_2, \text{constituent}(\text{right-hand}(p)))) \vee$
 $((CC_1 \in \text{constituent}(\text{left-hand}(p)) \vee$
 $CC_1 \in \text{constituent}(\text{right-hand}(p))) \rightarrow CC_1 = CC_2))$
 (2.4) os-sat(<OID, w_2 >, OS)が成立する。但し、

OS=OLかつOS=o-schema(CLASS(c-aspect(<OID, w_2 >)))である。

(2.5) $(C_1: I_1) \in W_1, (C_2: I_2) \in W_2$ であるとき、 C_1 と C_2 とが同一の祖先となるクラスで定義された属性attrを持つ場合、 I_1 と I_2 の属性attrは同一値を持つ。□

[定義6.2]の(2.1)は遷移規則が適用可能であることを、(2.2)は遷移規則適用後のaspect集合を規定する。(2.3)は、核aspectの移動に関する。(2.4)は、オブジェクトスキーマの制約条件の適用に関する。(2.5)は、属性値の伝播に関する。

aspect集合形式の集合上での直接の導出関係は、以下のように定義される。

[定義6.3] スキーマセットSSの下で、 $w_1 (\in \text{ASF}(\text{SS}, O_p))$ が $p(\in P)$ によって $w_2 (\in \text{ASF}(\text{SS}, O_p))$ に変更可能であるとき、 w_1 は w_2 を直接に導出(derive)するといつて、 $w_1 \Rightarrow w_2$ と記す。□

記号 $\Rightarrow$ は、オブジェクトのaspect集合形式の集合上の関係を表す。また、aspect集合形式の集合上での導出関係 $\Rightarrow$ は、以下のように定義される。

[定義6.4] スキーマセットSSの下で、 $\text{ASF}(\text{SS}, O_p)$ 上の関係 $\Rightarrow$ の反射的かつ推移的閉包を $\Rightarrow^*$ と記す。つまり、 $w (\in \text{ASF}(\text{SS}, O_p))$ に対して $w \Rightarrow^* w$ である。さらに、系列 $w_1, w_2, \dots, w_m (\in \text{ASF}(\text{SS}, O_p))$ があつて、 $w \Rightarrow w_1, w_1 \Rightarrow w_2, \dots, w_{m-1} \Rightarrow w_m$ という直接の導出の繰返しが存在する。このとき、 w は w_m を導出するという。□

$O_t = \langle \text{OID}, w_t \rangle, w_t \Rightarrow^* w_m, w_m \in \text{ASF}(\text{SS}, O_p)$ であるとき、[定義6.2]、[定義6.3]、[定義6.4]から w_t から w_m に属性値の伝播が行われる。これに対して、[2]では直接の導出における属性値の伝播しか扱っていないため、 w_t から w_m を導出するパスが複数存在するとき、 w_m の属性値の状態は導出パスに依存して異なり、かつ情報の喪失が起きる。

MAORIの操作言語では、aspect集合の操作には次の3つの関数が使われる。

(1) new(C) $\rightarrow$ OID

クラス名Cで示されるクラスが定義するaspectをただ1つ持つオブジェクトを生成し、そのオブジェクト識別子OIDを返す。このオブジェクトの核aspectは、唯一

のaspectである。

(2) delete(OID) $\rightarrow$ void

オブジェクト識別子OIDで示されるオブジェクトを削除する。

(3) migrate(OID, $\{S_1 [I, S_2, S_3, \dots]\}, \{[T_1, T_2, \dots]\}) \rightarrow$ void

オブジェクト識別子OIDで示されるオブジェクトに関して、 $\{S_1 [I, S_2, S_3, \dots]\}$ で示されるsource-aspect列から $\{[T_1, T_2, \dots]\}$ で示されるtarget-aspect列へのOMを行う。ここで、 $S_1, S_2, S_3, \dots, T_1, T_2, \dots$ の各々はクラス名であり、オブジェクトのaspectを指定する。但し、同一のクラス名の異なる生起は添字により区別され、異なるaspectを表わす。また、[...]は選択的な指定を表わす。

[定義6.5] スキーマセットSSの下で、時刻preceding(t)におけるオブジェクトOに対する関数migrate(OID, $\{S_1 [I, S_2, S_3, \dots]\}, \{[T_1, T_2, \dots]\})$ の実行は、次の(1)、(2)が成立するとき時刻におけるOMイベントeを生起する。但し、OIDはオブジェクトOの識別子である。また、 $\text{CS}(O_p) = (\text{CS}(O_{\text{preceding}(t)}) - \{S_1 [I, S_2, S_3, \dots]\}) \cup \{[T_1, T_2, \dots]\}$ である。

(1) $\{S_1 [I, S_2, S_3, \dots]\} \subseteq \text{CS}(O_{\text{preceding}(t)})$

(2) $\text{AS}(O_{\text{preceding}(t)}) \xrightarrow{*} \text{AS}(O_p)$ である。□

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けるOMの枠組みを提案した。この枠組みは、実世界において意味のない、あるいは誤ったOM生起を回避するための制御仕様としてオブジェクトスキーマと遷移規則を用いる。オブジェクトスキーマは静的制約の定義であり、オブジェクトの基本的な性質を表わす核aspectを通してオブジェクトに動的に関連付けられる。また、遷移規則は動的制約を定義する書換え規則であり、文献[2]の制御規則の形式より(1)条件付き、(2)文脈依存といった点で記述能力が高くなっている。

今後の課題としては、(1)OM管理機能の試作、(2)多面的オブジェクトの併合・分割モデルの検討、(3)版管理、スキーマ進化に対する多面的オブジェクトの応用の検討などを考えている。

参考文献

- [1] 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", 信学技報, DE95-73 (1995-12)
- [2] Qing L., Guozhu D.: "A framework for object migration in object-oriented databases", Data & Knowledge Eng., 13, pp.221-242(1994)
- [3] 福村晃夫, 稲垣康善: "オートマトン・形式言語理論と計算理論", 岩波講座・情報科学6, 岩波書店(1982)