Scenario/Role/Objectモデルによるトランザクション処理メカニズム

2K－9

片山　佳則
（株）富士通研究所 情報研

1．はじめに

オブジェクト指向プログラミングにおいて，実世界を実現するためには，トランザクション処理のための有効な方式を確立する必要がある．トランザクション処理とは，最終的には，いかにプログラマの負担を少なくして，実世界の動作をシミュレートさせるかに帰着する．オブジェクト指向プログラミングでは，トランザクション処理の実現に対して，メソッドの起動をコントロールすることが基本となる．この状況において，オブジェクト指向プログラミングにおけるトランザクション処理の困難さの要因として，スーパーバイザー的立場のものを置けないことを挙げることができる．特にトランザクション処理では，時間のメカニズムを取りまとめるためにこのような立場が必要になる．

本稿では，これらの立場のプログラミングをオブジェクトとの親和性を踏まえて扱えるような方式として，既に提案しているS-R-O方式が有効であることを示す．この方式では，時間のメカニズムのために，局面，場面，時刻の概念を取り込み，S-R-Oは，それらをオブジェクト指向プログラミングに積極的に取り込めるメカニズムを備えている．

2．S-R-Oモデルの概要

我々はオブジェクト指向プログラミングにおける幾つかの問題を解決するための新しいモデルとしてS-R-Oモデルを提案している[1]．これは，オブジェクト指向の考え方にシナリオとロールという概念を導入して拡張したモデルである．その基本的立場は，応用領域側の記述をシナリオとロールが受け持ち，共有領域側をオブジェクトが受け持つことで，応用に依存するプログラムと依存しないプログラムを積極的に分けることにある．応用領域側のシナリオでは，共有領域側で与えられるオブジェクトに必要な応用依存プログラムをロールとして割り振る(ロールアサインメント)．従って，オブジェクトは，割り振られるロールによって実行できる付加的メソッドが決められる．ロールやシナリオが各応用領域ごとの特殊なプログラムを受け持ち，オブジェクトが持つプログラムは，その応用領域に依存することなく広く利用できるものである．

A Transaction Machanism for S/R/O Model
Yoshinori Katayama
FUJITSU LAB. ISIS
9-3,Nakase 1, Mihama, Chiba 261,Japan

3．S-R-O方式によるトランザクション処理法

ここでは，S-R-Oモデルが持つシナリオ，ロール，オブジェクトを用いていかにしてトランザクション処理を実現できるかそのメカニズムを示す．基本となる考え方は，メソッドの起動のコントロールをオブジェクトへのダイナミックなロールの張り付けで実行することである．

S-R-O方式では，トランザクション処理を扱うために，次のようにオブジェクトの状態，局面，場面，時刻の概念を考える．オブジェクトの状態は，ある時刻のスナップショットとする．従ってオブジェクト群の状態は，時刻の前後関係によって，半順序で順序づけられる．これらの関係を動作の対象となるオブジェクト集合を{O1,O2,...,On},時刻の順序関係を$t0 \leq t1 \leq \ldots \leq tm$として表1に示す．時刻tmの時のオブジェクトOnの状態をOn(tm)で表している．

表1　オブジェクトの状態集合

	t0	t1	t2	t3
O1	O1(t0)	O1(t1)	O1(t2)	O1(t3)
O2	O2(t0)	O2(t1)	O2(t2)	O2(t3)
O3	O3(t0)	O3(t1)	O3(t2)	O3(t3)

この表において時刻tmの時のオブジェクト集合の状態セット{O1(tm),O2(tm),...,On(tm)}を局面と呼び，これらの局面の中で実世界との整合性が保証される局面を場面と呼ぶ．つまり，局面は時区間におけるオブジェクトの状態の集合である．このように，局面／場面／時刻を設定することで，トランザクション処理のための制約が(1)局面別制約(2)局面間制約(3)場面間制約の3つに分けられる．これらの制約を満たすメソッドの記述をそれぞれロールの記述として個々のオブジェクトの記述と区別し，局面ロールや場面ロールとして時刻の推移と共に必要なオブジェクトにダイナミックに割り振る(ダイナミックなロールアサインメント)．局面ロール／場面ロールは状況の理論における基本的構成要素の一つである時空位置における空間的／時間的に対応するロールとして捕えることができる．ダイナミックな割り振りはシナリオが実現する．このようなメカニズムによって，トランザクションの処理を実現する．

図1に示した歯車の噛み合わせを例にすると時刻t0はすべての歯車が初期状態で止っている状態である．この時，すべての歯車オブジェクトには後述の回転前

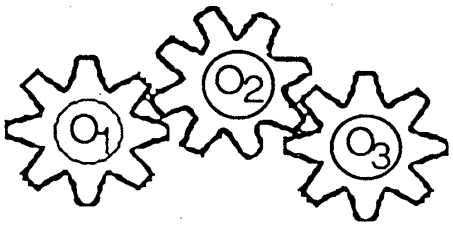

図1　歯車オブジェクト

ロールが割り振られている．時刻t1は，歯車O1が回転力を受けた（回転メッセージの起動）状態であり，歯車O1の状態が変化し隣接している歯車O2に回転力を伝える状態である．この時点で歯車O1は回転後ロールが割り振られる．時刻t2は，歯車O2が回転力を受けた（回転メッセージの起動）状態であり，歯車O2の状態が変化し隣接している歯車O3に回転力を伝える状態である．時刻t3は，歯車O3が回転力を受けた（回転メッセージの起動）状態であり，歯車O3の状態が変化した状態である．ここで，時刻t0とt3の状態集合は，それぞれ初期状態とすべての歯車が回転終了した状態であり，場面である．時刻t1, t2は局面である．各局面／場面において，歯車オブジェクトの果たすべきロールが異なる．

歯車の果たせるロールとして(a)回転前ロール：力を受けて隣接した歯車に力を伝える振る舞いを持つ(b)回転後ロール：力を受けない（受けても何もしない）を設定すると表１の時区間におけるロールの変化は表２になる．

表２　オブジェクトが果たすロールの変化
(a)：回転前ロール　(b)：回転後ロール

	t0	t1	t2	t3
O1	(a)	(b)	(b)	(b)
O2	(a)	(a)	(b)	(b)
O3	(a)	(a)	(a)	(b)

これらのロールの割り振りをS-R-O方式ではシナリオが管理する．つまり，各歯車オブジェクトはObjectユニットのオブジェクトとして共有情報により一般的に力を受けて回転するものとして実現されるが，この応用に用いているときには各時刻において果たすべきロールとしてRoleユニットで回転前／後のロール記述がなされ，シナリオがこれらのロールの間の制約や歯車オブジェクトへのダイナミックなロールの割り振りを管理する．

トランザクションの最小の単位は，一つの場面から次の場面までの間である．その他の局面は，部分動作となる．このようにしてトランザクション処理は，時刻と状態の関係から，局面と場面を切り分けて実現される．このトランザクション処理をオブジェクト指向プログラミングに素直に適用させる方式がロールアサインメントメカニズムを持ったS-R-O方式である．S-R-O方式では，シナリオがオブジェクトへのロールの割り振りを動的に行なうことで，通常の動作に対する制約を，状態変化と動作のタイミングとして与えられる時刻を考慮して局面内制約と局面間制約あるいは場面間制約を扱える．一般にこれらの制約をオブジェクト指向プログラミングで扱う場合は，各制約情報をオブジェクトの内部に取り込むかあるいはオブジェクトの外に置くかに分けられる．前者の場合は，状況に関わる外部の内部状態をオブジェクトの内部に取り込むことになる．これはオブジェクトとしては，インテリジェントなオブジェクトといえるがモジュラリティが悪化し，メンテナンスが困難になる．後者の場合は，全体を管理するオブジェクトを用意する必要があり，オブジェクト指向概念に新たな概念を導入する必要がある．ここで示したS-R-O方式はこの両方の側面を持つ．どちらの立場に分類されるかは，ロールが割り振られたオブジェクトをどのように解釈するかに依存する．つまり，ロールが割り振られたオブジェクトを一つのオブジェクトと見なすと前者のインテリジェントなオブジェクトを実現したことになり，ロールとオブジェクトを切り離して考えると，後者の立場になる．

特に重要なポイントは，シナリオで実施許可の制御を記述して管理することで，個々のオブジェクトにおける動作実行のメカニズムとそれらの管理のレベルが明確に分かれていることである．これが，トランザクションに関して通常のオブジェクト指向プログラミング以上に，さらにモジュラリティを上げる役割を果たしている．

4．まとめ

局面，場面，時刻の概念を取り込み，トランザクション処理を実現する方法としてS-R-Oモデルを捕え，その概要を示した．歯車の事例では，回転前／後のように状態主体でロールを与えている．実際の実装では，細かいロール分けがなされ，シナリオ内でロール間の制約や割り振りのデーモン等が与えられる．また，ロールの設定は，動作主体で行なうこと(歯車の場合は，駆動／受動などのロールの設定)も可能である．現在は，実装システムの実現だけでなくこれらのロール設定法や，ロールのView，複数ロールへの対処等幾つかの課題についての発展も進めている．

参考文献

[1] 片山，小林：" ソフトウェアとしての利用性を指向した新しいオブジェクトモデル(S-R-Oモデル) " Proc. Symposium of Information Science 1993, Jan 1993