3W－8

マルチメディアデータベースのためのオブジェクト処理系の構成方式

姚 左軍
東京大学

浜田 喬
学術情報センター

1　はじめに

単一のデータ構造で全てのデータを細かい部分まで管理する従来のデータベースの手法と違い、マルチメディアデータベースには、異種データの管理と処理について、各々の特定処理方法を用意することが必要である。それはデータベースの応用が益々広くなり、データ対象は益々複雑になるため、その対象を数字と文字列のような細かい部分まで細分して、単一のデータ構造で管理することは困難になるわけである。一種類のデータとそれに応じる特定の対処方法を組合せて、一つのミニシステム (a class of objects) を構成し、各種のオブジェクトをその種類ごとに幾つかのファイルにまとめ、集めてデータベースを作るのはマルチメディアデータベースの構成にとって一つの考えられる方法である。

オブジェクト指向データベースの概念 [1] により、オブジェクトの処理は完全にカプセルされ、全ての処理請求をメッセージの形式でオブジェクトの処理系に伝える。そして、オブジェクトの処理系がメッセージに応じで具体的な作業を行なう。そういう立場からオブジェクトの処理を考察すると、オブジェクトの処理過程は、1. 処理動作は外からのメッセージに応じて始まる、2. メッセージとその時の進行状態によって、処理系の一部を選び、実行する、3. 処理結果によって、オブジェクトの属性データあるいは示される内容を変える、という特徴がある。本稿は以上の特徴を基に、オブジェクトの処理系の各モジュールを実行中動的に組合せる方法について述べる。

2　FSM モデル

本案はオブジェクトの具体的な処理進行過程を予め予測せず、具体的な要求を受けた後、その要求とその時点の状態によってオブジェクトの処理系を動かし、その結果状態を変えるという手段で処理系を動的に調整する手法を採用する。

定義：任意のオブジェクトの処理系を次のような有限状態マシンにすることができる。

$$M = (S, E, \delta, T, s_0)$$

S：有限内部状態の空でない集合。

E：Mの全ての受理できるメッセージの集合。$E = E_{ex} \cup E_{in}$。E_{ex}は外からのメッセージの集合であり、E_{in} は内部メッセージの集合である。

δ：状態遷移関数。$S \times E \to S$

T：一つの外来メッセージを処理終った時、有限状態マシンが到着した状態 (待ち状態) の集合。$T \subseteq S$

s_0；初期状態。$s_0 \subseteq S$

オブジェクトの処理系は最初外からのメッセージにより立ち上げられる。オブジェクトの処理系が外からのメッセージを受けた後、場合によってその対処を多段階に分けて行なう。各段階をメッセージで接続する。即ち、前段階が終了後その処理結果による次の処理段階の選択情報を一つのメッセージにまとめ、処理系に伝える。このようなメッセージを内部メッセージと呼ぶ。

外からのメッセージを幾つかの段階に分けて処理する場合、その分ける基準は二つある。1. 前段階の処理が終って、次の処理段階に遷移する時、幾つかの可能な選択がある。2. 各処理段階で外からのメッセージに対する状態遷移の方向は一意である。内部メッセージで多方向の分岐選択を決める手法は、内部処理段階が多い場合、その処理系の組合せが論理上での複雑なトリー構造にならず、また外からのメッセージを処理しやすいという利点がある。

外からのメッセージの処理は最初メッセージの内容の分析から始まり、その後、前の段階の処理結

A Dynamical Method of Combination Object Processing System for Multi-media Database

Yao Zhuojun, University of Tokyo.
Takashi HAMADA, National Center for Science Information Systems.

果によって、適切なモジュールを選び次の段階に入り、過程を処理が完全に終了するまで繰り返す。こうすれば、途中の異なる処理段階で各種外からのメッセージを受けた場合でも、各々に正しい処理を行なうことが簡単に実現できる。一つの外からのメッセージの処理が完全に終了した後は、更に遷移を行なわなく、Mが最後の状態で次の外からのメッセージを待つ。即ち、待ち状態に入る。

3 FSMとDBS間のインタフェース

一つのデータベース中の全てのオブジェクトは各々独立にしている。処理系の起動、あるいは有限状態マシンの動作はDBSによって管理される。即ち、DBSが有限状態マシンの起動から完全に終了するまで、メッセージの受け方、関連のメソッドの実行、他のオブジェクトとの参照を協調管理する。図1は有限状態マシンとDBS間のインタフェースを示している。具体的な構成方法は次の通りである。

- 有限状態マシンの現在の状態、外からのメッセージ或は内部メッセージ、次の状態、及び処理関数の四つの要素を一つのレコードにまとめる。

  $$(cur\_state, message, next\_state, method)$$

  全てのレコードを合わせて一つの状態遷移表を構成する。

- DBSがオブジェクトの処理を始める時、状態遷移表(ST table)をロードする。

- 一様なデータ構造でメッセージを作り、イベント生成の方法でメッセージを発送する。

- DBSが全てのメッセージを収集し、状態遷移表を調べ、有限状態マシンを適切に動かす。

図1中のMessage QueueはDBSのもので、状態遷移表ST Tableは各オブジェクトの所有のものである。この表を有限状態マシンの起動と停止によって、作ったり、消したりして動的に変化させる。

4 おわりに

有限状態マシンのモデルを用いて、オブジェクトの処理系の各モジュールを結び繋ぎ、メッセージの形式を用いて、オブジェクト間の各種の関係を処理する。本案と文献[2]の案によって、全てのデータ

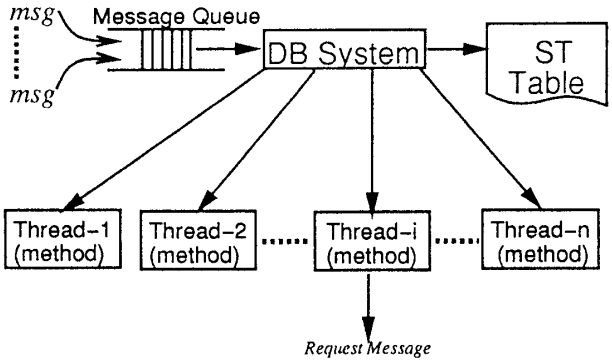

図1: 有限状態マシンとDBS間のインタフェース

と処理モジュール両方を持つオブジェクトを各々独立なデータにすることができる。またオブジェクト間の全ての意味関係を参照関係の対処のような形式で統一手法で対処できる。こうする利点は異種データを同一のデータベース中で管理する場合、その異種データ間にどのような論理関係があっても、独立に作ったオブジェクトを異なる時刻にデータベースに入れても、互いに“連絡”できるようになることである。これは大規模のマルチメディアデータベースの開発にとって極めて有利である。また、複雑な処理系に対して、1. 実際に多くの機能の内の一部が使われる時、どの機能を使うか、どのようにその機能を組合せるかを予め予測する必要はないし、複雑な判断はいらない、2. オブジェクトの処理系とDBS間のインターフェースが簡単になる、というメリットが得られる。そのため、オブジェクトの処理系のメンテナンス、拡張、再構成などの自由度を高めることができる。

参考文献

[1] W. Kim, J. F. Garza, M. Ballou and D. Woelk, “Architecture of the ORION Next-Generation Database System.”, *IEEE Transactions on Knowledge and Data Engineering.*, Vol.2 No.1 Mar. 1990.

[2] 姚左軍、濱田喬、“OODBにおけるクラス階層のインデックスの一案”、電子通信学会平成6年春季全国大会論文集