

データベースを基礎にした

2G-1 擬似オブジェクトベース機構の実現とシミュレーション

畠山正行 金子 勇

茨城大学

1 はじめに

対象世界のモデル化単位として「オブジェクト」の概念を用いてモデル化する方法は「現状では」最も自然なモデリング法の一つであろう。しかし、そのモデル表現を「変換」し続けて、シミュレーションに持ち込むまでの多くの過程において、ずっとオブジェクト単位のまま変換モデリングを続けて行くこと(つまり一貫モデリング)は現在ではまだ殆ど確立されていないといつてよいであろう。なぜならそのモデル表現をシミュレーションにまで持ち込む途上において、モデルの本質的な変性が生じるからである。つまりデータと操作が完全に分離されたプログラミング言語表現に「置換」されてしまうからである。これはモデリング過程の最終段階としてのシミュレーションには極めて都合が悪い。モデル化単位のオブジェクトがそのモデル化単位のままシミュレーションに持ち込める方式の方が、人間のモデリングに関する思考との一対一対応の観点からも、精密で一貫したモデリング論の観点からも、シミュレーションプログラムの構築や変更などの観点からなど全てに関して良い結果を生むことが十分合理的に予想出来る。その様なモデリング変換方法が確立されていない理由はデータとプログラムの永続的な共存格納管理の手法が確立されていなかったことにもあると考えられる。

シミュレーションに関するこのような基本的要求を満たすソリューションとして、我々はオブジェクト指向データベース(OODBMS)をプラットフォームとして、擬似オブジェクトベースと称する実行支援環境を構成、実装し、これを用いた数値シミュレーションを実現した。今回はその擬似オブジェクトベース機構の実現及び駆動の方法、及びその応用例について述べる。

2 オブジェクトベースとシミュレーション

オブジェクト指向モデリングによるモデル化とシミュレーションは次のような形になると考えている。通常はモデル化の結果としてクラス階層構造で書かれたプログラムが存在し、それをコンパイルして実行する。オブジェクト指向であるから、対象世界に新たな要素を加えたり、対象世界内オブジェクトにユーザから直接メッセージを送って、これを駆動することもできる。つまり、オブジェクト指向パラダイムはシミュレーション向きのモデリングパラダイムである事がわかる。

更に考察する。シミュレーションに必要な各種のパラメータや初期分布、境界条件、初期条件などについても本来はオブジェクト自身の属性値として内部に格納されているべきものであろう。また、生成される(通常大量の)データも本来はモデルオブジェクト内に一体的に管理される方が言うまでもなく適切であり、ユーザからみ

て便利でもある。従って、モデル化されたプログラムとデータはシステム内において、オブジェクトという単位でもって分離されず一体的に管理されるべきであるということになる。この場合、システム内に論理的にも物理的にも永続的かつ一体的に管理されているのが理想である。このシステム概念では、コンピュータ内に閉じ込められた対象世界はモデルオブジェクトを単位として格納、管理、取り出し、駆動、制御、変更、停止させられる。これはちょうどデータがこの様な扱いを受けるときにデータベースと称されるように、オブジェクトベースと称されるべきである、と考える。しかし、現状のUNIXシステムにおいてはプログラムの永続的な管理等はファイルシステムであり、従って、データベースで管理されるデータとは別々に管理されざるを得ない。この食い違いを解決する原理的な仕掛けは、ユーザの側からみれば、あたかも理想的なオブジェクトベースであるかのごとくに見えたり感じさせたりする様な環境を構築することである。この環境を擬似オブジェクトベースと称しても良いであろう。これが今回のテーマの前半である。このオブジェクトベースを駆動し、制御する仕掛けがオブジェクトベースシミュレーション支援環境である。我々は対象世界として流れの数値シミュレーション解析の世界を選びこのテーマを実現することを試み、擬似的なベース世界とその支援環境を構築した。

3 擬似オブジェクトベース機構とその実装

現状において、最もオブジェクトベースに近いシステムは、最近研究が盛んなオブジェクト指向データベース管理システム(OODBMS)であろう。しかしOODBMSは、あくまでデータベース管理システムであり、オブジェクト管理システムではない。そこで、このOODBMSに外部からリモートでオブジェクトへ直接操作するなどのオブジェクト管理機能を付加することによって、擬似的にオブジェクトベースを実現させることにする。

OODBMSとしてはONTOS^[1]を使用した。ONTOSはC++を親言語としており、オブジェクトの構造をC++のクラス定義の形で記述し、メソッドをC++のプログラムとして記述する。

ONTOSでの通常のアプリケーションは、データベース内のオブジェクトを、C++のプログラム上から操作するため、オブジェクトがユーザからは直接見えないようになっている。しかし今回、OODBをシミュレーションに用いようとしているため、データベース内のオブジェクトを外部から直接操作できることが望ましい。そこでONTOSの機能であるONTOS SQL^[2]を拡張し、E-OSQL(拡張ONTOS SQL)を製作した。

ONTOS SQLは主にデータベース内のオブジェクトを検索するために考え出されたものであるが、データベース内のオブジェクトのメソッドを起動することができる。E-OSQLはこれに、オブジェクトの生成、削除、データベースのオープン、クローズ、トランザクション制御などを加えたものである。表1に主なE-OSQLのコマンドを示す。

E-OSQLで行なえることはC++のプログラムからでも行なえるが、このE-OSQLコマンドは単なる文字列であるので、データベー

Realization and Simulation of the pseudo-Object-Base mechanism based on the OODBMS

Masayuki Hatakeyama, Isamu Kaneko
Ibaraki University

コマンド	機能
select	メソッド実行
insert	インスタンス生成
delete	インスタンス削除
class	クラス名検索
query	メソッド検索
transaction	トランザクション制御
commit	コミット
abort	アボート

表 1: E-OSQL コマンド

外から利用できる。そこで図1の様なクライアント-サーバ形式で利用する。

これにより、ネットワークを通してOODB内のオブジェクトにメッセージを送ってメソッドを実行することや、データベース内に新しいオブジェクトインスタンスを作成すること、データベースのトランザクションを制御することなどができるようになった。このような環境ではユーザはONTOS内にオブジェクトが存在するのと同じ操作環境におかれた状態になる。

4 数値風洞シミュレーションへの応用例

擬似オブジェクトベースの応用例として、直接シミュレーションモンテカルロ (DSMC) 法による希薄気体の数値シミュレーション (数値風洞) を取り上げた。DSMC法とは、気体分子の運動を確率的、統計的に直接シミュレーションする方法である。

まず、数値風洞をOMT法^[9]によりモデル表現し、それを擬似オブジェクトベース上で実装した。図2に今回の数値風洞のOMTオブジェクト図を示す。

実際に、E-OSQLで風洞を駆動している例を図3に示す。

数値風洞を擬似オブジェクトベース上で駆動することにより風洞の汎用的な利用が可能となる。今回の数値風洞においては、風洞内の境界の場所を変更させたり境界条件を変化させることなどが、従来のシミュレーションよりはるかに容易となったことを確認した。

5 終わりに

現状で最もオブジェクトベースに近いという理由によりOODBMSを採用して擬似オブジェクトベースを実現したが、シミュレーションの速度や計算できる規模などからすると、必ずしも最適なものとはいえないであろう。そこで次の段階として、OODBMSをベースにした擬似的なものではなく、より本格的なオブジェクトベースを構築する予定である。

引用文献

- [1] "ONTOS Developer's Guide", ONTOS, inc., 1991.
- [2] "ONTOS SQL Guide", ONTOS, inc., 1991.
- [3] J. ランボー 他著, 羽生田栄一 監訳, 「オブジェクト指向方法論 OMT」, 1992年, (株)トッパン.

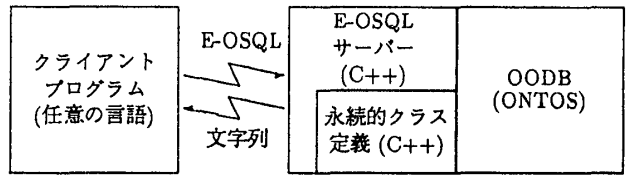


図 1: E-OSQL による OODB の利用形態

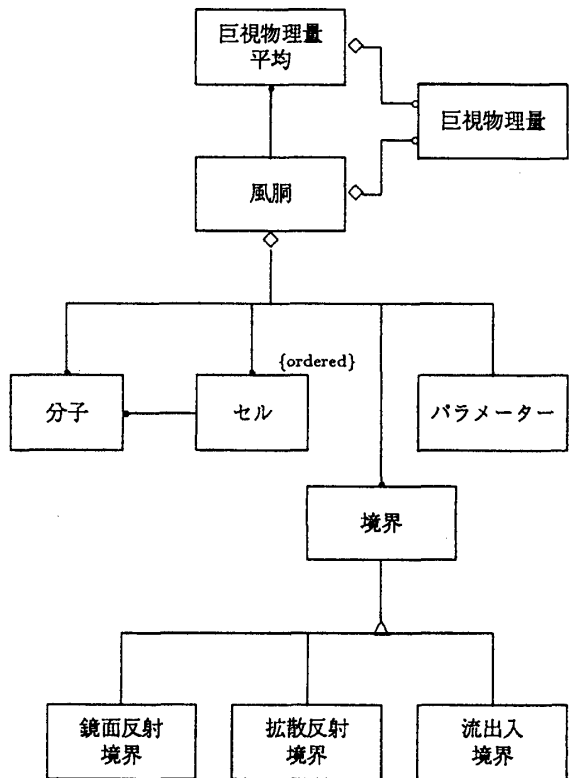


図 2: 数値風洞の OMT オブジェクト図

```
SQL> insert into Mirror_boundary(0,0,0,1,1,0.2,7,0,0);
#1Mirror_boundary
SQL> select0 add_boundary(#1Mirror_boundary)
2> from Wind_tunnel;
SQL> select0 process(0.1) from Wind_tunnel;
SQL> select0 macro_mean(0.1) from Wind_tunnel;
SQL> select0 macro().output() from Wind_tunnel;
field
30 15 15
1074 489 0 2 1535
1027 466 0 -4 1444
```

図 3: E-OSQL による風洞駆動例