

解説



シミュレーション技術の最近の動向

2.7 オブジェクト指向型離散系シミュレーション言語 MODSIM II[†]

(原題) MODSIM II : An object-oriented language for discrete-event simulation

J. D. Salt^{††} 甲斐宗徳(翻訳)^{†††}

1. はじめに

MODSIM II は、Modula と似たシンタックスリを持つ、コンパイル方式によるオブジェクト指向型の高次プログラミング言語であり、米国カリフォルニア州の CACI 社により製作・販売されている。MODSIM II は、会話型のグラフィックスやアニメーション機能を持つ離散イベントシミュレーションモデルを構築するためのものである²⁾。本稿では、MODSIM II の能力について技術的な点を簡単に解説する。

2. MODSIM II の開発に関する主な経緯

MODSIM II は Modsim という言語の改訂版であるが、これは最初米国国防総省のオブジェクト指向型の離散イベントシミュレーション言語の要求に応じて開発されたものである。最初の Modsim は並列処理マシンの上で動くように設計された。MODSIM II ではそのかわりに様々なポピュラーな商用マシンで実行できるようにした。MODSIM II は元の言語を多くの点で強化したものであり、とりわけ SIMGRAPHICS II を用いたグラフィックス能力の付加は大きい。本稿では MODSIM II のリリース 1.8 について述べる。

3. MODSIM II システム

MODSIM II システムは以下の要素から構成されている。

1. MODSIM II スマートコンパイラ mscomp
2. MODSIM II 標準ライブラリ
3. SIMGRAPHICS II グラフィカルエディタ

simdraw

4. SIMGRAPHICS II プレイバックユーティリティ simvideo

5. MODSIM II クラスブラウザ obj.mgr

このシステムの PC 版にはウィンドウベースの ModBench という開発環境が含まれている。さらに MODSIM II の全バージョンとも多数のサンプルプログラムが同梱されており、ユーザの調べたいときに、グラフィックスを用いたサンプルでもあるいは用いないサンプルでも参照できるようになっている。

MODSIM II コンパイラは中間コードとして C のコードを生成し、それがコンパイル、リンクされて、実行形式ができあがる。中間ステップとして C を用いたのは、MODSIM II に高いポータビリティを与えるためである。

4. 離散イベントシミュレーションの要件

Mitrani³⁾ は、良いシミュレーション言語とは、次の 5 つのカテゴリのファシリティを提供するものであると述べている。

1. エンティティおよびエンティティ集合に対する操作
2. 時間とイベントの取扱い
3. 疑似乱数の生成
4. 統計量の取得
5. 数値計算

MODSIM II はこれらの要件をすべて満たしており、以下にそれぞれの詳細について述べる。

MODSIM II におけるエンティティはオブジェクトにより表現される。オブジェクト指向パラダイムについての詳細はもう少し後で述べることにする。ここでは MODSIM II が、オブジェクトクラスの定義、生成、消滅、そしてそれらのクラスをスタックやキュー、ランク付け集合としてプ

[†] MODSIM II : An Object-oriented Language for Discrete-event Simulation by J. D. Salt (CACI PRODUCTS Division).

^{††} CACI Products Division
^{†††} 成蹊大学工学部経営工学科

プログラマに提供することができるということだけ述べておくことにする。

MODSIM II シミュレーションモデルマネージャは、シミュレーションクロックを管理し、イベントリストを処理する。シミュレーション時間は無次元の実数型変数であり、プログラマはシミュレーション時間の1単位が実時間のどのくらいの長さになるかを決めなければならない。MODSIM II は、プロセス中心のモデル構成概念をサポートしている。これはプログラマが任意にWAIT ステートメントを使うことにより、オブジェクトの完全な動きをコード化できることを意味している。このとき、各WAITの最初と最後にイベントに関する個別のプロシージャを書く必要はない。(訳者注: シミュレーションのモデルを構成するには、大別してプロセス中心のモデル構成概念かイベント中心のモデル構成概念が用いられる。イベントは、その中に時間経過を含まない事象であり、プロセスは互いに関係のあるイベントを時間順に並べ、その間に時間経過を含むようなものである。どちらを中心にモデル構成を行うかにより前述の2つの概念の区別が行われるが、時間経過が含まれないプロセスがイベントであるとも考えることも可能であり、プロセスはイベントを含む概念と見ることもできる。) Law と Kelton⁴⁾ は、このプロセス中心のモデル構成概念を用いたアプローチは、イベント中心のモデル構成概念を用いたアプローチに比べてずっと自然であり、コード量も少なく済むと考えている。Hills⁵⁾ は、プロセス中心のモデル構成概念を用いてシミュレーションモデルを記述するのは、イベント中心のモデル構成概念を用いて記述するよりも難しいと主張しているが、前者の方が結果的により小さく速いコードになるとしている。他のシミュレーション言語と比べてユニークなところは、MODSIM II は、1つのオブジェクトに、同時に複数の動き (activities) を持つことを可能にしている点である。

オブジェクトに複数のインスタンスが存在し、各々がシミュレーション時刻で複数のタスクを実行できる能力があるということは、MODSIM II はプログラマに対して、オブジェクト間でアクションの同期をとらせる適切な手段を提供しなければならないということの意味する。MODSIM II

ライブラリは、オブジェクトの時間を消費するメソッドに割込みをかけるプロシージャを提供する。MODSIM II のシンタックスでは、WAIT ステートメントに“ON INTERRUPT”という記述を付加することができる。これは、メソッドが割込みのシグナルを受け取り、そのあとに行う振舞いを指定するためのものである。MODSIM II のメソッドは、同じオブジェクトのメソッドや他のオブジェクトのメソッドの終了を待つことができる。また、それ以外に TriggerObj というオブジェクトの発火を待つこともできる。このオブジェクトは、それを待つすべてのメソッドのための開始シグナルとしての機能を果たす。最後に、ResourceObj というクラスのオブジェクトがあり、あるリソースが利用可能になるのを待つメソッドをブロックする機構を作るために使われるものである。

MODSIM II の標準ライブラリで利用できるあるクラスのライブラリには、Lehmer の乗算合同法を用いた疑似乱数列を生成するものがある。あらかじめ定義された乱数の生成の種は10種類提供されており、それらはSIMSCRIPT II、5の言語と同じものである。もし10種類の独立な乱数列では不十分であるという場合でも、別の乱数の種を探すのは簡単なことである。乱数を生成するアルゴリズムは、 $2^{31}-1$ 個の乱数を生成するまで同じ乱数列が繰り返されることはない。

シミュレーションを行おうとするユーザは、たいてい、作成したシステム中のキューの長さや待ち時間についての統計量を求めたいと望むことが多い。キューについての統計量に関しては、MODSIM II はプログラムの実行中に自動的に統計量をとることのできるスタックやキュー、ランク付き集合というクラスを提供してくれる。すでに述べたように、リクエストを満足するのに十分なリソースが利用可能になるまでそのサービスのリクエストをブロックするような、有限のリソースとして使用されるクラスがある。このクラスのオブジェクトは、サービスを待つオブジェクトのキューについての統計量を集計することもできる。待ち時間に関する統計量については、プログラマは、オブジェクトのライフサイクルの中の適切な時点でシミュレーションクロックの現在時刻を記録するわずかなコードを書かなければなら

ない。

MODSIM IIの数値計算ファシリティとしては、汎用の高水準言語と同様に四則演算、指数関数、対数関数、三角関数といったものが用意されている。これは、数値計算を実行するには外部のルーチン呼び出すことを必要とする他の多くのシミュレーション言語と対照的な部分である。

5. オブジェクト指向

オブジェクト指向は、もともとはシミュレーションの研究者達によって開発されたものである。そこで、MODSIM IIでもオブジェクト指向パラダイムはフルサポートされている。プログラムはオブジェクトのクラスを定義し、実行時にそれらのクラスのインスタンスをダイナミックに作り出すことができる。オブジェクトというものは、1つのプログラムをコードとデータに厳密に分離することを取りやめたものである。オブジェクトは、それ自身のフィールドに、レコードと同様な方法でデータを内包する。オブジェクトはまた、そのオブジェクトが実行することのできるオペレーション集合であるメソッドも内包している。オブジェクトのフィールドは、そのメソッド以外のメソッドによっては変更されることはできない。したがって、あるクラスのために定義されたメソッドは、そのクラスが外界とどのように相互作用するかを完全に指定している。MODSIM IIは、データの内包について最新のアプローチをとっており、オブジェクトのフィールドは、それを所有するオブジェクト以外のオブジェクトからは、読むことは許可されているが書き込むことはできない。これは、フィールドを読むためだけメソッドが使われなければならないという Smalltalk とは対照的である。もし、フィールドやメソッドへの外部アクセスを制限したければ、それらは private で宣言されればよい。

MODSIM IIは、オブジェクトのクラスが持つメソッドを3種類に分類している。ASK METHOD は、同時に、すなわちシミュレーションにおける時間の経過がまったくなく、実行される。TELL METHOD と WAITFOR METHOD は、両方とも実行中にシミュレーション時間が経過することになり、それらのメソッドは WAIT ステートメントを含むことがで

きる。

オブジェクトのクラスは他のクラスからの継承をすることができる。これは、サブクラスがスーパークラスのすべてのフィールドとすべてのメソッドを継承することを意味する。MODSIM IIは多重継承を許可しており、複数のスーパークラスから継承するためのクラスがあることを意味する。MODSIM IIはまた、反復継承もサポートしている。これはサブクラスが1つの親 (ancestor) クラスから複数のルートによって継承を行うものである。多重継承によって起こるかもしれないフィールド名やメソッド名のいかなる衝突も回避できるように言語のメカニズムが構築されている。

スーパークラスから継承されたメソッドは、継承するクラスの中でその実現を変更することができる。これは、MODSIM IIではクラスの定義において OVERRIDE というキーワードを用いることによって行うことができる。通常は、オーバーライドされたメソッドは、継承された振舞いを拡張することになる。元のメソッドの振舞いをオーバーライドされたメソッドに編入するのは簡単なことである。異なるクラスのオブジェクトに同じ名前を持たせる能力は、多形性 (polymorphism) として知られている。

MODSIM IIにおける特別な種類のオブジェクトクラスは、PROTO クラスである。PROTO クラスは、キューやスタックのようなグループのクラスであり、指定されたクラスの要素に限られたメンバシップを持つことができる。これは Eiffel のパラメータ化されたクラスの考えに相当するものである。多くのオブジェクト指向言語では欠けていることだが、継承によって全体の構成を再利用したいならば、この能力は必要である。

6. MODSIM IIの他の能力

最新のソフトウェア工学の実践に従って、MODSIM IIは、強い型付けとモジュール性のコンセプトを含んでいる。単純なデータ型としては、BOOLEAN, INTEGER, REAL, STRING, およびユーザ定義の列挙 (範囲) 型がある。STRING 型は完全に動的なものである。構造化されたデータ型には、ARRAY, RECORD, そしてもちろん OBJECT がある。ある型から別の型へのキャストは、陽に実

行されなければならない、最も頻繁に必要とされる変換を実行するために、いくつかの関数が用意されている。MODSIM IIの数値型はすべて倍精度であるので、特に倍精度の数値型という区別をする必要はない。構造化プログラミングで使用するループ構造もすべて利用可能であるが、GO TOステートメントはない。

MODSIM IIのプログラムはどれもちょうど1つのmainモジュールを持たなければならない。mainモジュールはそれ1つでプログラム全体であってもよい。通常は、mainモジュールは小さく、プログラムの残りの部分は多数のモジュールのペアから成っている。MODSIM IIのモジュール構造は、Adaと似ており、定義部を実行部の記述から分離している。各モジュールのペアは、定義モジュールと実行モジュールから成っている。定義モジュールは型定義（クラスの定義も含む）、定数の定義、変数の宣言を含んでいるが、実行可能なコードは含んでいない。実行モジュールは、定義モジュールとマッチするように宣言されたメソッドとプロシージャのソースコードを含んでいる。MODSIM IIにはグローバルデータというものはない。モジュールの外側から定義や宣言を見えるようにするには、プログラマはIMPORTステートメントを使って陽に制御することになる。この規則の唯一の例外は、定義モジュールのすべては、それとペアを組む実行モジュールからは自動的に見えるようになっていることである。

MODSIM IIにおけるモジュールの依存性は複雑になるかもしれないので、MODSIM IIシステムはmscompとスマートコンパイラを持っている。これはモジュール間のインポートパターンを追跡し、編集されたモジュールとそれに依存するモジュールだけが再コンパイルされる。巡回するような依存関係は自動的に処理されることができる。もし望むならばモジュールのコンパイルの順序を制御するためのファシリティもあるが、プログラマにとってmakeファイルを書くという必要は取り除かれている。

コアとなるMODSIM II言語は小さいがエレガントなもので、ISOレベル0のPascalにその能力は近づいている。この言語の真のパワーはMODSIM IIと一緒に提供される標準ライブラリ

にある。標準ライブラリには、ファイル入出力、離散イベントシミュレーションメカニズム、グループクラス、乱数、対話型動画グラフィックスなどのためのファシリティが含まれている。そのグラフィックライブラリを用いることにより、プログラマはユーザの入力をグラフィカルに処理することができ、シミュレーション結果を絵で表現することもできる。とりわけシステムのダイナミックな振舞いを示すアニメーションを生成することができるようにしたことが重要である。MODSIM IIオブジェクトの1つのクラスは1つのウィンドウを表し、したがって1つのMODSIM IIプログラムはプログラマが望むだけの数のウィンドウを制御することができる。

MODSIM IIのグラフィックスインタフェースはメニューやダイアログボックスのクラスを持っている。ダイアログボックスはテキストや数値の表示、ボタン、ラジオボタン、リストボックスを含んだ形をしている。MODSIM IIはホストのオペレーティングシステムにあるウィンドウシステムを利用するので、これらのグラフィック用オブジェクトは、そのウィンドウシステムで用いられているものと同じ形状のものが用いられる。したがって、MODSIM IIプログラムを、たとえばWindows 3.1.1からSun Open Windowsに移植するときにも、何の書き直しも必要としない。ソースコードは同じで、必要になるのは再コンパイルすることだけである。

プレゼンテーション用グラフィックスは、グラフ、ヒストグラム、パイチャートを描くことができ、それぞれ最大100個のデータセットについて表示することができる。これらのグラフィックスは、シミュレーションが実行している間にダイナミックに更新されることができる。MODSIM IIのグラフは、現在の設定範囲を超える値をプロットする必要が出てきたときには、自動的にスケールを合わせてくれる。レベルメータはある変数のレベルを表すことができ、クロックは現在のシミュレーション時刻を表示するのに使われる。

アニメーションに使われるグラフィックのオブジェクトもシミュレーションの実行中に、いろいろ変換されたり、回転、スケールを合わせ直したりすることができる。もし望むならば、プログラマによって指定された座標系で、これらの動きを

実行することができる。MODSIM IIのウィンドウオブジェクトは、プログラマに、ウィンドウ表示を好きな範囲にズームインさせたり、ウィンドウサイズより広いエリアを左右にパンさせたりできる。アニメーションを対話的にすることも簡単である。どのグラフィックのオブジェクトも BeSelected というメソッドを持っていて、マウスを使ってそのオブジェクトが選択されたときにはいつでもこのメソッドが呼び出される。BeSelected メソッドをオーバーライドすることによって、プログラマはユーザとの対話を開始することができる。たとえば、ダイアログボックスを出してそのオブジェクトのフィールドの現在の値を表示させ、ユーザにその値を編集させることができる。これにより、シミュレーションの実行後に受動的にアニメーションを示すのではなく、シミュレーションの実行中にユーザが対話することができるモデルを書くことができる。

MODSIM IIのグラフィックスは、SIMGRAPHICS IIのグラフィエディタである simdraw を使って非常に簡単に作ることができる。マウスを使うグラフィックインタフェースによって、1行のコードを書くこともなく、あらゆる種類のグラフィックスを作ることが可能である。インタフェースとプレゼンテーション用のグラフィックスは、プログラマのアプリケーションに自由に合わせて作ることができる。アイコンはボックスやサークルのような基本的な形状から作り出すことができ、いろいろな複雑な絵を作り出すためにイメージ用のツリー型メニューの中にそのアイコンを登録することができる。simdraw は、作り出された画像をグラフィックライブラリに保存することができる。MODSIM IIのプログラムからそれらを利用することができる。simdraw はまた、背景として使用するラスタグラフィックを、適切なフォーマットで取り込むこともできる。

MODSIM IIのプログラムで出力されるアニメーションは、simvideo というユーティリティを使用して、録画したり再生したりすることができる。このユーティリティは simpost と simdump というユーティリティも含んでいる。simpost は SIMGRAPHICS IIの画像をポストスクリプトフォーマットに変換するもので、MODSIM IIプログラムで作られた画像はどんなものでもポストス

クリプト対応プリンタで印刷することができる。simdump はドキュメンテーションの手助けをするもので、グラフィックライブラリにあるグラフィックオブジェクトの構造の一覧をテキストで与えてくれるものである。

7. MODSIM IIの現在の状況

MODSIM IIは様々な企業でのシミュレーションのプロジェクトでうまく使われているようである。Royal Ordnance 社は対戦車用の地雷原の効果モデル化するためにこれを使った。Euro-tunnel 社は、トンネルのターミナルで道路の交通量をシミュレートするのに使っている。Euro-control 社は MODSIM II で大規模なモデルを構築し、大気制御問題の研究に使っている。Royal Navy 社のオペレーショナル解析のスタッフは、海戦用のモデリングに使っている。

CACI 社は、ネットワークシステムをモデル化するためのフレームワークである SIMOBJECT を開発するために MODSIM II を使用してきた。この開発から、通信ネットワーク用のシミュレーションツールとして、COMNET III が開発されている。(COMNET IIIは連載解説の方で紹介する予定である。)

MODSIM II は UNIX や PC など幅広いプラットフォーム上で使用できる。PC ユーザは Windows 版あるいは OS/2 版を選ぶことができるが、使用に際してはマウスと適当な C コンパイラが必要である。MODSIM II の UNIX 版は、Sun SPARC, HP 700 シリーズ, IBM RS/6000 といったワークステーション用が用意されている。

CACI 社は米国ワシントン DC と La Jolla において5日間のトレーニングコース、英国 Camberley とオランダの Maastricht では3日間のコースを開設している。価格や試用版についての詳細は CACI 社で受け付けている。

参考文献

- 1) MODSIM II Reference Manual, CACI Products Company, La Jolla (Jan. 1993).
- 2) SIMGRAPHICS II User's Manual for MODSIM II, CACI Products Company, La Jolla (Jan. 1993).
- 3) Mitrani, I.: Simulation Techniques for Discrete-Event Systems, Cambridge University Press, Cambridge (1982).
- 4) Averill, M. Law and David Kelton, W.: Simulation modeling & Analysis, 2nd edition, MacGraw-Hill, New York (1991).
- 5) Hills, P. R.: An Introduction to Simulation Using Simula, Norwegian Computing Centre, Oslo (1973).

(平成7年6月6日受付)

J. D. Salt

英国 Exeter 州立大学にてロシア語とフランス語の文学士を取得, その後英国 Newcastle-upon-Tyne 市立大学にて計算機科学の理学修士を取得. 最近では Brunel University にて, シミュレーションのモデリングに関する PhD を取得するべく離散イベントシミュレーションの図式化について研究している. 1988年1月に, Hunting Engineering Limited (HEL) に入り, 英国国防省向けの飛行編隊のシミュレーションモデルを作成. その後 Eurotunnel 社に入社し MODSIM II を利用する. そして, CACI 社に移籍して製作部門の主任技術者となり, SIMSCRIPT II.5 および MODSIM II のトレーナーとしてヨーロッパ各地においてセミナーを開催している.

