3Q-9

C言語を用いたシミュレーション言語実行プログラム

一宮　和喜**　岩本　真治*

*日本電気（株）C&Cシステムインタフェース技術本部、　**日本電気技術情報システム開発（株）

1，はじめに

現在、様々なシステムにおける性能が問題になり、システムの性能を予測することが必要になっている。システムの性能予測にはおもに数値解析による方法とシミュレーションによる方法などがある。しかし、システムが複雑になると数値解析による方法がむずかしくなる。また、既存のシミュレーション言語ではシステムが複雑になっていくとシミュレートの速度が遅く計算機のメモリを使いすぎるなどの問題がでてくる。そこで複雑なシステムを記述できるできるシミュレーション言語とその実行プログラムが必要になった。

報告者らは、新しいシミュレーション言語「シムシム」を開発し、前記の問題を解決しようと試みた。本稿ではシミュレーション言語「シムシム」と実行プログラムの概要について報告する。

2，シミュレーション言語「シムシム」

2．1，シミュレーション言語「シムシム」の特徴

シミュレーション言語「シムシム」は以下のような特徴がある。

○離散型シミュレーションである。

シミュレーションには連続型シミュレーションと離散型シミュレーションがある。連続型シミュレーションでは連続的に変化する量をシミュレートする。たとえば、電圧、水位、位置等を連続して変化する量とし、変化の様子を微分方程式でモデル化し、シミュレートする。離散型シミュレーションは待ち行列で表わされるようなモデルを扱う。シムシムは離散型のシミュレーションであり、人、データのひとまとまりなどのもの（以後、トランザクションと呼ぶ）のシステム内の待ち、移動などをシミュレートする。

○もの中心のモデル化。

シミュレーション言語にはもの中心のモデル化と事象中心のモデル化がある。もの中心のモデル化は、モデル化しようとするシステム内をトランザクションがどのように動くかを記述する。それに対して、事象中心のモデル化は事象の発生により、システムがどう変化するかを記述する。シムシムはトランザクションのフローを記述することにより、シミュレートするモデルを定義するもの中心のシミュレーション言語である。

○Cコンパイラを使っている。

GPSSやSLAMは離散型のもの中心のモデル化ができるシミュレーション言語である。キューやトランザクション発生をするブロックを組み合わせることによりモデルを簡単に定義できる。しかもキューやトランザクション発生をするブロックにいろいろな機能があり、簡単にシミュレーションの結果を得ることができる。しかし、複雑なシステムを細かく記述し、モデル化するときはFORTRANのサブルーチンをリンクするという方法を取らなければならない。Cコンパイラを使ったシミュレーション言語シムシムはC言語と同じ形の式をモデルに直接書き込むことにより、細かい記述を簡単にモデルに書き込んで実行することができる。式には四則演算等の算術演算子や関係演算子、論理演算子、型変換、インクリメントとデクリメント演算子、ビット演算子、代入演算子、条件式がつかえる。

また、シムシムの構文はC言語に似ている。それゆえC言語のプログラムの経験がある人にとってシムシムは入りやすいシミュレーション言語である。

○機能を単純化した。

シムシムはキューやトランザクション発生のブロックの機能を落として、単純化することにより、シミュレーションにかかる時間を短縮することができた。必要があれば落とした機能はC言語と同じ形の式を書きこむことにより実現できるので、それぞれの機能を単純化し自由に制限なく組み合わせられるようにした。

2．2，シムシムの記述方法

シミュレーション言語「シムシム」はC言語に似た記述ができる。

シミュレーションソースファイルの例を以下に示す。

```
var MAB[2];
var JIKAN;
model{
        xvar x[10];
        gener(rand()%40);
        x[3]=10;
        x[4]=0;
        x[8]=time;
        x[2]=1;
        goto XABA;
        gener(rand()%40);
        x[3]=10;
        x[4]=1;
        x[8]=time;
        x[2]=0;
        goto XABB;

XABA:   hold(6);
        que[0 == MAB[x[4]],0];
        MAB[x[4]] = MAB[x[4]] + 1;
        goto BFB;
BDBA:   if(x[2] != 0)goto BDBB;
        MAB[x[4]] = MAB[x[4]] - 1;
        goto YAAA;
XABB:   hold(6);
        que[0 == MAB[x[4]],0];
        MAB[x[4]] = MAB[x[4]] + 1;
        goto BFB;
BDBB:   if(x[2] != 1)goto BDB0;
        MAB[x[4]] = MAB[x[4]] - 1;
        goto YAAB;
BFB:    hold(x[3]);
        goto BDBA;
BDB0:   printf("x[2]=%d, x[4]=%d¥n", x[2], x[4]);
        down(1);

YAAA:   JIKAN += time - x[8];
        term(1);
YAAB:   JIKAN += time - x[8];
        term(1);
}

control{
        int i;
        for(i = 0;i < 10;i++){
          JIKAN = 0;
          srand(i);
          start(1000);
          printf("goukei = %d      heikin = %f ¥n"
                 , JIKAN, (float)JIKAN/(float)1000);
        }
}
```

シムシムのシミュレーションソースファイルはモデル部とコントロール部に分かれている。モデル部はト

Simulation Language Executing Program by C-language

Kazuki ICHIMIYA**, Shinji IWAMOTO*

*NEC Corp. **NEC Scientific Information System Development,Ltd.

ランザクションの流れを定義し、コントロール部は初期値の設定、シミュレーションの実行、結果出力をプログラムしている。モデル部とコントロール部はともに構文的にはＣ言語と同様であるが、モデル部は複数のトランザクションが同時にフローの中を移動するモデルを定義してあるので、シミュレート時の動作の仕方はＣ言語とは異なる。

２．２．１，モデル部

モデル部では複数のものがモデルの中を移動していくフローの定義をする。

トランザクションのフローはＣ言語プログラムの処理フローに似ている。if文で分岐し、goto文でラベルまでジャンプする。しかし、Ｃ言語とは異なり、トランザクションがgener文で発生し、term文で消去され、また、hold文 によって、あるシミュレーション時刻までトランザクションを止めておくことができる。

モデル部の文について説明する。

○ｘｖａｒ文

トランザクションの属性を表わすトランザクション変数を宣言する。

○ｇｅｎｅｒ文

トランザクションの生成を行なう。

○ｉｆ文

後ろに続く条件により、トランザクションの流れを分岐させる。Ｃ言語と同様な記述ができる。

○ｇｏｔｏ文

トランザクションをラベルの位置まで移動させるのに用いる。

○ｈｏｌｄ文

トランザクションを引数で表わされた時間（シミュレーション時間）だけ留めておく。

○ｑｕｅ文

トランザクションを引数の条件式が真になるまで留めておく。

○ｓｐｌｉｔ文

トランザクションが通過する毎に新しいトランザクションをラベルの位置に発生させる。

○ｔｅｒｍ文

トランザクションを消す。引数の値を累計していきコントロール部の start()の引数に達した時点でシミュレーションを終了する。

○ｄｏｗｎ文

シミュレーションを終了する。

ラベル名と変数名はシステムで予約している名前と重ならない限り、英文字から始まる英数字のみ文字列が使える。

２．２．２，コントロール部

コントロール部は初期値の設定、シミュレーションの実行、結果出力をプログラムしている。コントロール部はＣ言語と同様な記述ができる。

start文によりシミュレーションを実行し、トランザクションが通過したときのterm文の引数の値の累計がstart文の引数に達した場合、シミュレーションを終了する。

結果出力はＣ言語のprintf()等の関数が使用できる。

２．２．３，環境変数

変数には、シミュレーションの全体で使える環境変数と、トランザクションの状態を表わすトランザクション変数がある。トランザクション変数はモデル部の中でのみ用いることができる。

３，プログラム構成

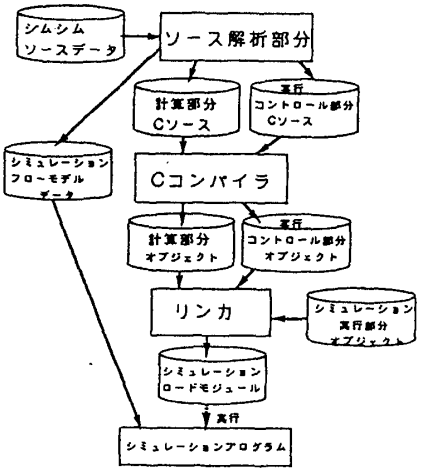
シミュレーション言語実行プログラムは、Ｃコンパイラを利用して実行される。

まず、ソース解析部が、シミュレーションのモデルを定義したソースファイルを読み込み、シミュレーションフローモデルデータ、計算部分のＣ言語ソースファイル、及び、シミュレーション実行コントロール部分のＣ言語ソースファイルを生成する。

次に生成された二つのＣ言語ソースファイルをコンパイルし、シミュレーション実行部分オブジェクトとリンクして、シミュレーションロードモジュールを作成する。

そして、ロードモジュールは実行され、シミュレーションフローモデルデータを読み込み、シミュレーションを実行する。

下図はシミュレーションの全体の構成を示したものである。



４，実行速度

シムシムをＳＬＡＭとＧＰＳＳと比較するために、同じモデルでシミュレーションを実行させて時間を計測した。その結果、シムシムはシミュレーションソースデータを読み込んで実行までの時間が他のシミュレーション言語よりかかったが、シミュレーションを実行している時間が他のシミュレーション言語よりかなり短かった。

５，おわりに

Ｃコンパイラを用いたシミュレーション言語「シムシム」実行プログラムの開発を報告した。

今後このシミュレーション言語を使っていく過程で使いやすいように改良していく。

＜参考文献＞

[1]森戸　晋、相沢りえ子　共著　ＳＬＡＭによるシステム・シミュレーション入門