

オブジェクト指向を取り入れたシミュレーションシステム*

2Q-5

塩沢 そのみ† 澤田 健彦‡ 岩本 真治†

日本電気(株) インタフェース技術研究所†
 日本電気(株) C&C オープンシステム技術本部‡

1 はじめに

ネットワーク等の複雑なシステムのシミュレーションをFORTRANやC言語などの汎用言語で行なうと、一般に記述が複雑になり、プログラミングの工数も大きくなる。また、シミュレーション言語を用いることによって、効率よくシミュレーションを実施することが可能になるが、中にはプログラムのモジュール化が困難な言語があったり、対象とするシステムによっては、かえってモデル化が難しくなり、煩雑なプログラミングが必要となることがある。そこで私たちは、これらの問題を解決するために、モデル化が直感的で分かりやすく、プログラムのモジュール化が可能で、さらにプログラミングが容易なシミュレーション言語の開発を目指している。

現在プロトタイプが完成し、実際に実システムのシミュレーションに用いて評価しているところである。ここでは、その評価について報告する。始めに、このシミュレーション言語の概要を述べ、シミュレーションの対象となったシステムを簡単に説明する。次にモデル化、プログラミング、トランザクションの記述について、上で取り上げた問題点がどのように解決されたのかを述べ、最後に、この言語の特徴の一つであるシミュレーションの実行を制御する機能について説明する。

2 シミュレーション言語の概要

離散型のシミュレーション言語には、モデル化するシステムの状態に着目して、その変化を記述する方法と、システム内の要素の動きに着目してその動作を記述する方法がある。ここで述べるシミュレーション言語は、後者のモデル化の方法を用いる。また、この言語はオブジェクト指向の概念を取り入れており、C++に類似した記述を行なう。すなわち、オブジェクト内でのみ有効な内部静的変数の宣言を行ない、メソッドを記述する。このオブジェクトはトランザクション(オブジェクト指向では、メッセージに相当する)を受信することにより起動され、トランザクションを送受信することにより情報交換を行なう。

*A simulation system adopting object-oriented concept, Sonomi SHIOZAWA, Takehiko SAWADA, and Shinji IWAMOTO

†Interface Eng. Labs., NEC Corporation

‡C&C Open Systems Eng. Div., NEC Corporation

3 対象としたシステムの概要

対象としたシステムは2種類の通信路(LAN, SCSI)と、4種類のノード(NodeA ~ NodeD)により、図1に示すように構成されている。

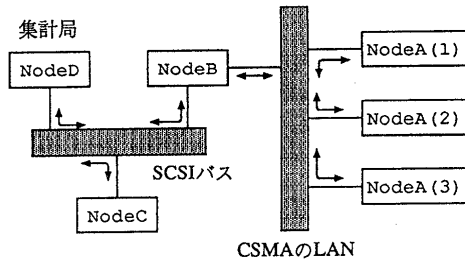


図1: システムの構成

また、このシステム中を流れる通信データは各NodeA, NodeCから発生し、集計局NodeDに流れる。このとき、NodeAから発生したものはNodeBを経由する。

4 モデル化について

上記のシステムのモデルは、図2に示すように、システ

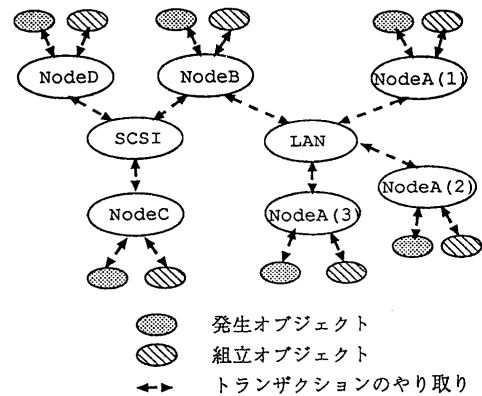


図2: オブジェクトの構成

ムの物理的構成要素(NodeA ~ NodeD, SCSI, LANなど)と、機能(トランザクションの発生や、分割されたトランザクションを組立等)を表すオブジェクトから構成され

る。このように、モデルのオブジェクトと実システムの構成要素や機能を一対一に対応させることが可能であり、分かりやすいモデルを作るのに非常に役立った。また、それぞれのオブジェクトは独立したものとして定義されるので、モデルの再利用も容易である。ここでは以前に作成したLANのモデル(約120ラインのプログラム)を利用し、コーディングの手間を省いた。

5 プログラミングについて

クラス継承

一般のオブジェクト指向言語と同様に、クラス継承を用いたコーディングができる。そのため、コーディング量は少なく済み、プログラムの変更も用意に行なうことができた。

C言語を用いた記述

C言語の演算文、制御文などが使用でき、また確率分布関数などのシミュレーションに有用なライブラリを備えている。このため、細かい動作の記述も自由に行なうことができた。

キュークラス

キューイングモデルを基本とするシミュレーションで、必須となるFIFOのモデルがキュークラスとして準備されている。

トランザクションの分割

この言語では、プロトコルのデータ分割をモデル化するときなどに、有用な制御文が用意されている。それは、同じ属性値のトランザクションを新しく生成するため、分割されたどちらのトランザクションも分割前の情報を知ることができる。

変数の記述

オブジェクト内の変数は、受信するトランザクション毎に初期化される変数と、そのオブジェクトが生成された時に初期化される変数がある。このような変数の使い分けを行なうことにより、トランザクションの伝達時間の累計など、シミュレーションの結果出力に関する記述が容易であった。

6 トランザクションの記述について

プログラム上のトランザクションのやり取りが、全て実際のトランザクションの流れを表しているわけではない。プログラミング上、実際のトランザクションの流れをはずれて、処理を記述する必要が出てくる。しかし、このような場合でも、図3に示すように、サブルーチン的なトランザクションの制御を行なうメソッド(この言語での表記は“<%”を用いて示す)を用いることにより、実際のトランザクションの流れに近い記述ができる。そのため、プログラムの構造にとらわれることなく、実際のトランザクションの流れをイメージ通りにプログラミングできた。

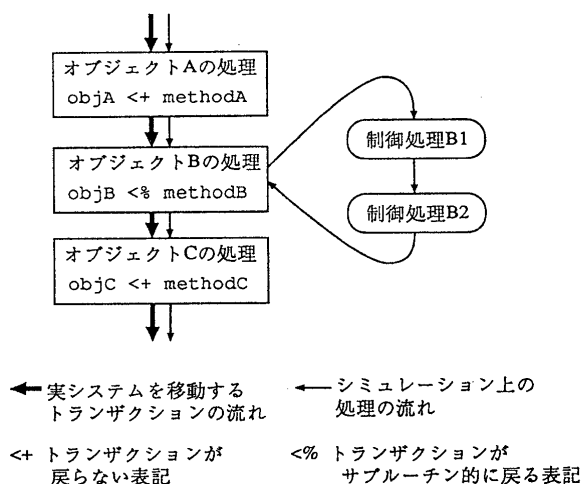


図3: トランザクションの流れ

7 並列動作の確認

シミュレーションプログラムが正しく動作していることを確認することは、シミュレーション作業の中でも難しい作業の一つである。特に、ネットワークのシミュレーションのように、同時にネットワーク上を動き回るトランザクションが存在し、この状態を表現している場合など、本当にそのトランザクションが正しく動作しているのかを確認する作業は大変難しい。そこで、このシミュレーション言語には、シミュレーションの実行中に、外部から与えるコマンドによって、その動作を制御する機能を実現している。この機能を用いることにより、シミュレーションの中断/再開、シミュレーション実行中にノードやトランザクションが有している変数の参照/変更が可能になった。この機能は、今回のシミュレーションにおいてもプログラムの動作の確認に非常に有効であった。

8 終わりに

以上述べてきたように、現在開発しているシミュレーション言語を実際のシミュレーションに適用させた結果、分かりやすいモデル化、プログラミングのモジュール化、プログラミングの工数の削減という点について、非常に有効であったと考えている。今後は実用システムとして、さらに開発を進めていく予定である。

9 参考文献

- [1] 一宮, 岩本: C言語を用いたシミュレーション言語実行プログラム, 情処全大第39回予稿集, II-1319/1320 (1989)