

# 通信網擬似システムにおける自動実行機能の実現

4Bb-6

久保田 稔\* 藤原 信一\*\* 中村 元紀\*

NTTネットワークサービスシステム研究所\* 株式会社SRA\*\*

## 1. はじめに

通信ソフトウェアのプロトタイプ開発環境として、通信網擬似システムを構築した [1]。本通信ソフトウェアはオブジェクト指向モデルに基づく分散プログラムであり、単一ノード上で実行されるプログラムと比べて機能の検証やデバッグを行うことが困難である。そこで機能検証の自動化を主たる目的として、通信網擬似システム上に自動実行機能を構築した。本稿ではこの実現手法を中心に述べる。

## 2. 通信網擬似システム

本稿で対象とする通信網擬似システム（図1）は、大きく分けて、実行管理プロセス、通信サーバ、仮想的ネットワーク構成要素（ノード、伝送路、スイッチ、端末）からなる。これらはUNIXプロセスにより実現されている。仮想ノードは、本ノード上で実行される複数のアプリケーション（通信サービスソフトウェア）からなる。各アプリケーションは複数のオブジェクトからなり、一つのUNIXプロセスとして実行される。実行管理プロセスは、仮想ネットワーク構成要素を実現しているUNIXプロセスの実行を制御する。通信サーバは、仮想ノード上のアプリケーション間のメッセージと後述するイベントの送受を行う。

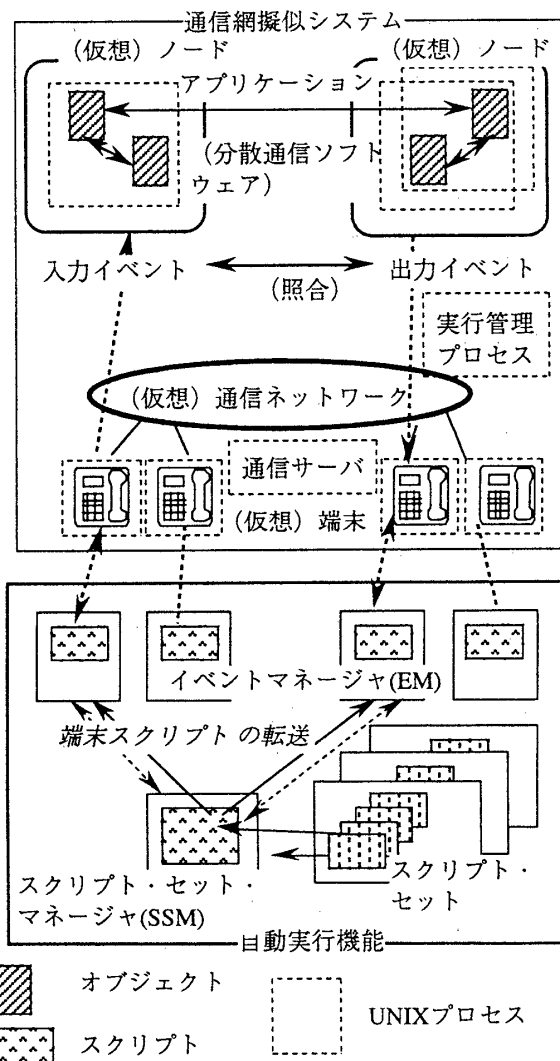
## 3. 自動実行機能の基本機構

分散通信ソフトウェアの実行は、外部からイベント（入力イベント）を入力し、それに対し実行結果（出力イベント）が出力されるものとする。自動実行機能は、あらかじめ定められた手順（スクリプト）にしたがって、(1) 入力イベントの生成、と (2) 出力イベントの監視、を必要ならば同期をとりながら自動的に行う。通信網擬似システムにおいては、入力イベントの

生成は仮想端末の操作により行う。

通信ソフトウェアの機能検証を行うために、これに関わる端末の処理の制御を行うスクリプト（端末スクリプト）と、端末スクリプトの実行を制御する手順（実行制御スクリプト）の組が必要となる。これをスクリプト・セットと呼ぶ。端末の処理は、入力イベントの生成と出力イベントの受け付けを意味する。

自動実行機能により機能検証が自動的に行われるので、オペレータによる操作誤りの可能性がなくなり、対象プログラムの誤りの再現性が高まる。これは機能



Implementation of Automatic Execution Control Facilities for Telecommunication Network Simulation System

Minoru KUBOTA\*, Shin'ichi FUJIWARA\*\*, and Motonori NAKAMURA\*

\*NTT Network Service Systems Laboratories, \*\*SRA Inc.

図1 通信網擬似システムと自動実行機能

検証の作業の効率を高める。さらに入力試験イベントに対応する出力イベントが自動実行機能により自動的にチェックできるため、機能追加時に分散通信ソフトウェアの機能検証を繰り返すことが容易になる。

#### 4. 自動実行機能の要求条件

##### (1) スクリプト間の同期

分散通信ソフトウェアの実行の再現性を高めるため、異なる端末から送出されるイベントの順序制御が必要になる。このため、スクリプト間で同期をとる必要がある。

##### (2) 複数スクリプト・セットの同時実行

複数の通信サービスが同時に実行する際の競合に関する誤りの検出を容易にするため、複数のスクリプト・セットを同時に実行できる必要がある。

##### (3) スクリプトの再利用

通信ソフトウェアでは検証すべき項目が多数あり、機能追加・変更に伴う検証作業も多いので、スクリプトを変更せずに再利用できることが望ましい。

##### (4) 実行経過の記録

機能検証を容易にするため、実行経過の記録をできるだけ多く保存する必要がある。

#### 5. 自動実行機能の実現

##### 5.1 基本機構

自動実行機能は、複数のイベント・マネージャ (EM) とスクリプト・セット・マネージャ (SSM) により実現され (図 1)、これらは同期をとりながらイベントの送受を制御する。EM は端末上で端末スクリプトを解釈実行する。SSM は実行制御プロセス上で実行制御スクリプトを解釈実行する。分散通信ソフトウェアからのイベントは、端末によって処理される。

通信網擬似システムは X ウィンドウシステム上に構築されており、スクリプトの命令の文法および実行機構は Tcl/Tk を用いて実現した。Tcl 本来の命令の他に自動実行機能のための独自の命令を追加した。

##### 5.2 実現方法

###### (1) スクリプト間の同期

一つのスクリプト・セットに属するスクリプトが互いに実行順序を制御するための、同期及び実行のスピードを制御する命令を新たに実現した。同期の基本機構は、一つのスクリプトが他スクリプトからの同期シグナルが送られてくることを待つ、ことにより実現される。また端末スクリプトでは、スクリプトの実行

タイミングを制御するために、イベントを待つ命令がある。またスクリプト実行のエラー検出のため、同期処理や信号を待っている時のタイムアウト機構をサポートしている。

###### (2) スクリプト・セットの同時実行

スクリプトに記述された命令は、X ウィンドウシステムにタイムアウト関数を登録することにより一定時間毎に呼び出される Tcl インタプリタによって一行毎に実行される。各々のスクリプトには、それぞれ一つのインタプリタが割り当てられるとともに、複数のスクリプト・セットを同時に実行することができる。

###### (3) スクリプトの再利用性

スクリプト・セットに含まれる端末スクリプトはパラメータ化されて選択可能である。すなわち、スクリプト・セット中の実行制御スクリプトが使用する端末スクリプトは、スクリプト起動時に容易に変更可能である。また端末スクリプトは実行開始時に端末へ転送されるが、この際パラメータの指定により端末スクリプトの動作の変更 (たとえば接続先の端末の番号の変更) を行える。これらにより、一つのスクリプト・セットを用いて、実行条件を変えながら複数の機能の検証を行うことが容易にできる。

###### (4) 実行経過の記録

各スクリプト・セットの実行の記録は、ログとして Tcl インタプリタが実行管理プロセスに送る。実行管理プロセスでは、ログの生成元の端末やログの種類毎に表示/非表示を指定することにより、実行経過の解析を容易にしている。

#### 4. まとめ

自動操作機能により、分散通信ソフトウェアの機能検証が自動的に行われるので、オペレータによる操作誤りがなくなり、試験対象プログラムの誤りの再現性が高まる。これは機能検証の効率を高める。さらに入力イベントに対応する出力イベントが自動操作機能により自動的にチェックできるため、機能検証を繰り返すことが容易になる。本稿で述べたシステムは UNIX ワークステーション上で稼働しており、分散通信サービスソフトウェアのプロトタイプ開発に用いている。

#### 参考文献

- [1] M. Kubota and K. Maruyama : Testbed for Distributed Object-Oriented Telecommunication Service Software, JC-CNSS '94, pp. 210-215 (1994)