マルチメディア通信と分散処理 37 － 1
（１９８８．５．２０）

セションシミュレータによる通信ソフトウェアの開発

勝山 光太郎　中川路 哲男　宮内 直人　水野 忠則

三菱電機㈱　情報電子研究所

ＯＳＩの規約に基づく通信ソフトウェアの開発機会が増大してきている。特に高位層のプロトコルを実装するソフトウェアを効率的に開発していく必要がある。そのために我々は、セション層のサービスを模擬するセションシミュレータを開発した。本報告では、シミュレータの機能、構成について述べるとともに、シミュレータを利用した試験支援系についてふれている。また、シミュレータを利用したソフトウェアの開発についても考察を加えた。

A development of communication software using the session-simulator

Kotaro Katsuyama　Tetsuo Nakakawaji　Naoto Miyauchi　Tadanori Mizuno

Information Systems and Electronics Development Laboratory

MITSUBISHI ELECTRIC CORPORATION

5-1-1 OFUNA KAMAKURA-CITY KANAGAWA 247 JAPAN

The number of case to develop communication software based on OSI standards is increasing. In particular, it is required that communication software for higher layer protocol is developed effectively. Therefore we developed the session-simulator which simulates services of the session layer. In this paper, we present functions and structure of the session-simulator, and discuss the testing environment using the session-simulator.

１．はじめに

情報通信システムの発展に伴い、各種通信ソフトウェアの開発機会が増すとともに、その規模も増大してきている。また、情報通信システムにおいて、各種の情報処理装置を相互に接続する必要性から、ＯＳＩ（開放型システム間相互接続）〔1〕に関する標準化がＩＳＯ／ＣＣＩＴＴで進められ、その実現が急務となってきた。

このため、通信ソフトウェアを効率的に開発する必要性が高まり、通信ソフトウェア開発のための体系的な開発支援系の研究〔2,3〕や、通信ソフトウェア開発支援のための各種ソフトウェアパッケージの開発〔4〕が盛んになっている。

また特に、ＯＳＩでの規格化もセション層までは既に国際規格となっており、プレゼンテーション層、応用層でも国際規格ができつつある。今後、続々と応用層の規格化が進むと予想される。

このような状況において、応用層プロトコルを実現するためのソフトウェアを，効率よく開発することが重要な課題となる。

そこで、我々はＯＳＩ基本参照モデルの階層化の原則を考慮し、セション層のサービス〔5〕を提供するセションシミュレータを、オブジェクト指向言語superC〔6〕を用いて開発した。

本報告では、セションシミュレータの機能、構成、このシミュレータを利用した試験支援系、及びシミュレータを利用したソフトウェアの開発について報告する。

以下、２章では本シミュレータを開発する要因となった通信ソフトウェア開発上の課題について述べ、３章ではそれらの課題を克服するために設けたシミュレータ設計方針について触れる。４章ではシミュレータの機能を詳述し、５章ではオブジェクト指向言語によって実現されるソフトウェアの構成、及び動作の概要を述べ、そして６章ではこのシミュレータを利用した試験支援系について提案する。７章ではこのシュミレータを利用したソフトウェアの開発について考察する。

２．通信ソフトウェア開発上の課題

通信ソフトウェアは基本的に汎用ソフトウェアと同一の性格を有しているが、開発にあたって次のような課題が存在する。

（１）移植性

相互に通信するソフトウェアを複数かつ異なった機種上に実装する場合が多い。また、実装する時期も逐次的に行なうのではなく、同時に並行して行なう必要がある。このため、開発するソフトウェアの機種間での移植性が課題となる。

（２）マルチプロセス

通信処理は本質的に非同期な処理を含んでいる。そのため、複数のプロセスを並列的に処理する必要があり、マルチプロセス機能が必須となる。従って、それに合わせた環境整備が課題となる。

（３）応答性

通信処理では、一定時間内での応答が保証されている必要があり、この応答性の確保が課題となる。

（４）試験環境

従来のプロトコルの試験では、通信回線を流れるデータをトレースしたり、プロトコルアナライザを使用しデータをチェックするといった方法があった。実際に通信を行う試験では、設備の準備や設定が困難なことや試験効率が悪いといった問題がある。従って、これらを克服し効率的に試験を行なうことが課題となる。

３．セションシミュレータの設計方針

２章で述べた課題を考慮し、次の設計方針とした。

（１）移植性に対して

開発するソフトウェアのセション層インタフェースが、セションシミュレータのものとターゲットマシンのものと同一となるようにする。

セションインタフェースの変更に容易に対応できるようにセションインタフェースライブラリとシミュレータ本体を明確に分けて設計する。

（2）マルチプロセスに対して

ｕｎｉｘの本来もつマルチプロセス機能を利用する。プロセス間通信機能を利用し通信をシミュレートする。

（3）応答性に対して

開発する応用層のソフトウェアについて、セション層への通信要求のイベントをトレースし、時間を測定できるようにし、所望の応答性能が得られるか確認できるようにする。

（4）試験環境に対して

本シミュレータの第一の狙いはこの試験環境に関する課題を克服する点にある。以下の指針を設定した。

①実際の通信では試験困難な箇所も試験できるようにする。

②意図的に異常状態を起せるようにする。

③マルチウインドウ環境を利用し、通信している様子を表示し、分かり易い環境を提供する。

4．セションシミュレータの機能

4．1　機能

3章で述べた設計方針に基づき、セションシミュレータの主な機能として、以下の機能を実現した。

（1）機能単位の選択機能

セション層では、カーネル、全2重、半2重、小同期、大同期、アクティビティ管理等の機能単位を定義しており、コネクション確立時に、機能単位を折衝することができる。セションシミュレータでは、機能単位を予め設定しておくことによって、この折衝機能の確認ができる。

機能単位の設定は、機能単位を登録してあるファイルに対する更新によって実現する。これをサポートするユーティリティを設けた。

セションシミュレータは、起動時にこのファイルを参照し機能単位を設定する。

（2）トレースログ機能

セションサービスプリミティブのやりとりをトレースし、逐次表示したり、ファイルとしてログをとることができるようにした。

（3）逐次及び連続実行機能

セションサービスプリミティブのやりとり毎にパラメータや利用者データを確認しながら実行できる逐次モードと、連続的に実行する連続モードを選択できるようにした。

逐次モードはシステム試験の初期の段階で有効であり、連続モードはヒートラン試験等に有効である。

（4）異常シーケンスの発生機能

データを意図的に欠落させることによって、次のような異常シーケンスを発生できるようにした。

①同期点番号の異常

②タイムアウト

③トークン制御の異常

（5）セションインタフェースライブラリ

インタフェースライブラリとしてＣ言語の関数形式のものを用意した。

セションサービスプリミティブとは、基本的に1対1に対応している。

（6）コネクションレスセションサービスの提供

コネクションレスセションサービスを提供するようにした。

4．2　ユーザインタフェース

ユーザインタフェースは、マルチウインドウ環境としX-windowを利用し実現した。

メニューウインドウからセションシミュレータを起動する。起動時の初期画面を図1に示す。

逐次実行と連続実行はそれぞれモードのメニューから選択する。図1では連続実行のモードが選択されている。

また、逐次実行モードでは、"FORWARD" ,"BACKWARD" をそれぞれ選択することで、次画面、前画面をみることを可能とした。

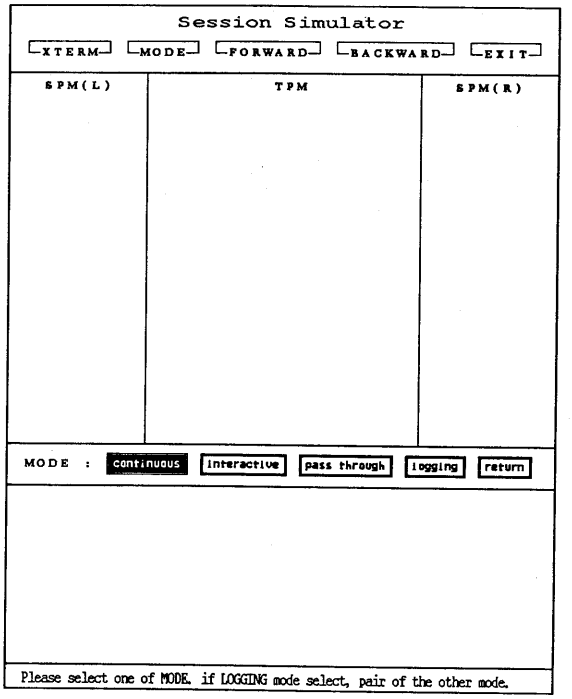

図1　セションシミュレータの表示例（初期画面）

"XTERM" はターミナルエミュレータを起動する。
"EXIT"はセションシミュレータを終了する。
サービスプリビティブのやり取りの様子を図２に示すような形で表示するようにした。
シーケンス図の形式で表示することによって分かり易いものとした。図中のプリミティブ名をマウスでクリックしprint を指定することで、パラメータの内容、dumpでデータの内容、listでパラメータの説明、timeでプリミティブを受け付けた時刻を表示するようにした。

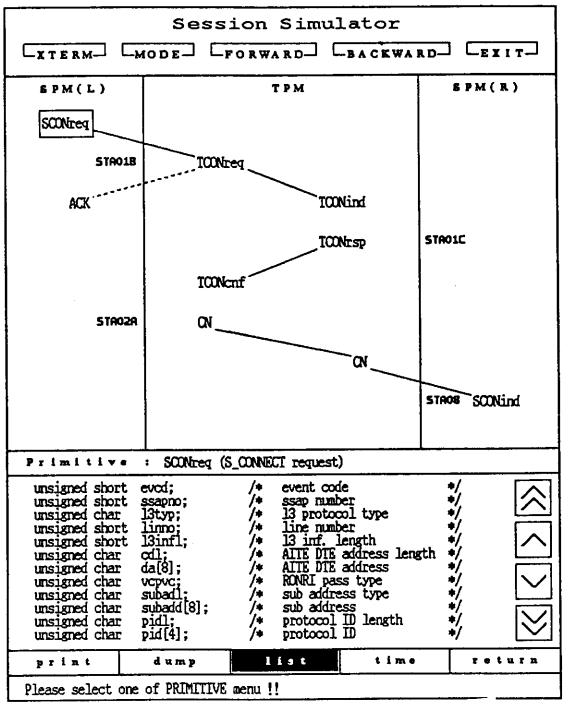

図２　セションシミュレータの表示例

５．セションシミュレータの構成と動作概要

５．１　構成

本シミュレータはオブジェクト指向言語superCを用いて開発した。
シミュレータを構成する主なオブジェクトについてその機能概要を述べる。

①socket
上位層及び下位層とのサービスプリミティブのやりとりと、コネクションの管理をするオブジェクトである。

②connect
コネクションの数だけ存在し、コネクションに関する情報を持つオブジェクトである。コネクションの確立および解放にともなって、socketより生成および消去が行なわれる。

③spm
セションプロトコルマシンを実現するオブジェクトである。状態遷移表に記述されているプロトコル処理を実現する。

④tpm
トランスポート層以下のデータの流れをシミュレートするオブジェクトである。エラー処理の試験のためにデータの欠落などもこの中で意図的に行なうことができる。

⑤sta
ｓｐｍの状態を表わすオブジェクトである。セション層の状態が遷移する度に生成および消去される。

⑥msg
セションシミュレータ内の各オブジェクト間でメソッドの引数として受け渡しされるデータそのものを表わすオブジェクトである。

⑦window
ウインドウの情報を管理するオブジェクトである。ウインドウの表示、ウインドウへの表示制御を行なう

⑧menu
メニューの情報を管理するオブジェクトである。ポップアップメニュー等のメニューの制御を行なう。

５．２　動作概要

ここでは、シミュレータ内の主なオブジェクトの生成とオブジェクト間のメッセージオブジェクトのやりとりについて、コネクション確立時を例に用いて説明する。（図３参照）。○内の番号は図３中の番号に対応している。

①シミュレータは起動されると同時にユーザプログラムとのプロセス間通信路の設定とｓｏｃｋｅｔの生成を行う。ここではプロセス間通信路として、ｕｎｉｘ４．２ＢＳＤのソケット機構を使用している。
②ユーザプロセスがシミュレータに対してコネクション確立要求（ＳＣＯＮｒｅｑ）を発行し、ｓｏｃｋｅｔがそれを受け取る。
③ｓｏｃｋｅｔはｃｏｎｎｅｃｔとそのサービスプリミティブに対応したｍｓｇを生成して、ｃｏｎｎｅｃｔにそのｍｓｇを送る。ｃｏｎｎｅｃｔには識別子がふられ、以降ｓｏｃｋｅｔはそれによってユーザプロセスからのデータを渡すｃｏｎｎｅｃｔを識別する。
④ｃｏｎｎｅｃｔは生成されると、ｔｐｍと２つのｓｐｍを生成して、片方のｓｐｍにｍｓｇを送る。
⑤ｓｐｍは生成されると、自分の初期状態に対応したｓｔａ（ｓｔａ０１）を生成し、ｓｔａ０１にｍｓｇを送る。

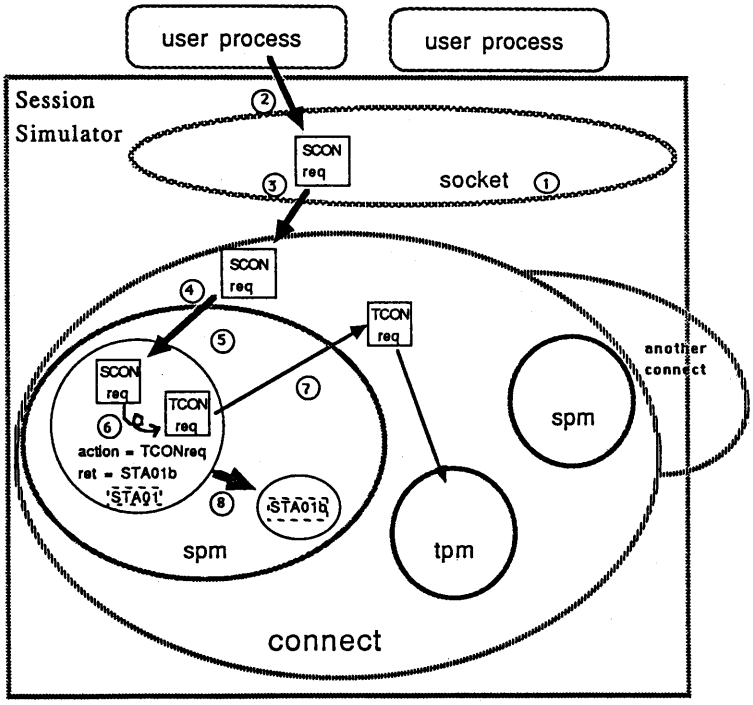

図３　動作概要

⑧生成されたｓｔａ０１は発行条件を確認し、受け取ったｍｓｇからとるべきアクション（ＴＣＯＮｒｅｑの送信）を引いてそれをｍｓｇとし、次の状態（ｓｔａ０１ｂ）を返す。
⑦ｓｐｍはそのｍｓｇをｔｐｍに送る。
⑧同時にｓｐｍはｓｔａ０１を消去し、ｓｔａ０１ｂを生成する。

６．セションシミュレータを利用した試験支援系

通信ソフトウェアの開発の幾つかのフェーズの中でセションシミュレータは試験のフェーズで利用されるものである。ここでは、セションシミュレータを利用した試験支援系について述べる。

試験支援系の論理構成を図４に示す。ＯＳＩコンフォーマンス試験の手法と枠組み[8]で示されている外部試験（External Testing）の論理的構成を１つの計算機上で実現した。

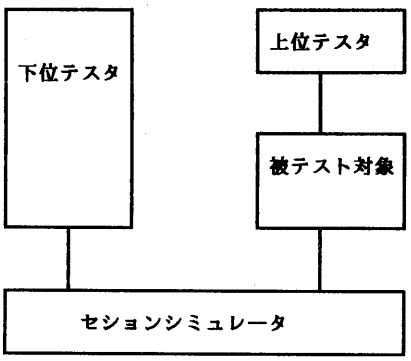

図４　試験支援系の論理構成

ここでは、ＦＴＡＭの試験を例にとり、外部試験の３つの方法について、実施方法を示す。

（１）遠隔試験法

ＦＴＡＭの応答側の試験に利用する。上位テスタは存在しない。下位テスタから、プレゼンテーション層、ＡＣＳＥのプロトコルデータにマッピングしたＦＴＡＭプロトコルデータをセションシミュレータを利用してＦＴＡＭ応答側に送り、動作を試験する。

（２）分散試験法

ＦＴＡＭの起動側の試験に利用する。この場合は上位テスタを開発する必要がある。テストコーディネーションのやり方は、いろいろ考えられるが、オペレータあるいは、ファイルからのテストイベントの入力によりテストを実行することになる。

（３）協調試験法

ＦＴＡＭ起動側の試験に利用する。上位テスタと下位テスタの間で、テスト管理プロトコルを決める必要がある。テスト管理プロトコルデータ単位を通すチャネルとしてセションシミュレータのコネクションレスサービスを利用する方法が有効である。

それぞれ起動側応答側独立に試験した後に、図５に示すように実際に起動側と応答側で通信する方法がある。テスタが存在しない場合もこの方法となる。

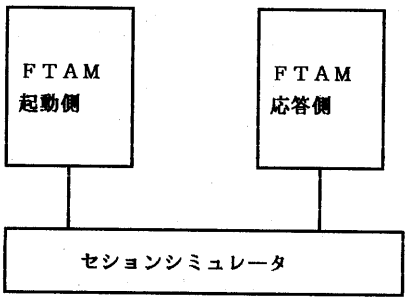

図５　テスト構成

７．考察

セションシミュレータを利用した通信ソフトウェアの開発に関して、ここでは、ＦＴＡＭ開発[8]の経験を基に考察を加える。

７．１　移植性

シミュレータを利用したソフトウェアの開発の目的は、開発したソフトウェアの各機種への展開にある。

移植上の問題として次の３つがある。

①言語

②ＯＳインタフェース

③プログラムインタフェース

これらの問題に対する解決策を以下に述べる。

（１）言語

一般に言われることであるが、高級言語を使用する必要がある。

我々は、Ｃ言語を拡張したオブジェクト指向言語superCを使用することとした。

（２）ＯＳインタフェース

ターゲット機のＯＳとシミュレータの動作するｕｎｉｘインタフェースの差をなるべく局所化し、移植を容易にする必要がある。

通信プログラムは複雑な画面制御を行なうプログラムなどとは違って、それ程複雑なＯＳの機能を使用する訳ではない、特に高位層のプロトコルのソフトウェアはＨ／Ｗに依存した処理もないために、どの機種のＯＳにもある基本的な機能を使用するだけで記述可能である。

我々は、これらのＯＳの基本機能を仮想ＯＳインタフェースという形で共通的に定義することによって、ソフトウェアの移植性の向上を図った。

仮想ＯＳの概念図を図６に示す。

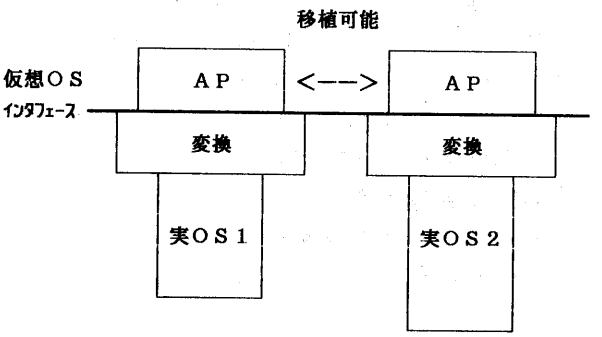

図６　仮想ＯＳの概念図

（3）プログラムインタフェース

ターゲット機のセションインタフェースとシミュレータのセションインタフェース（プログラムインタフェース）を同一にする。

7．2　マルチプロセス

ｕｎｉｘ４．２ＢＳＤのもつプロセス間通信機能であるソケットを利用しシミュレータとユーザプロセスの間のデータのやり取りを行なった。

オブジェクト指向アプローチで設計しデータをオブジェクトとしたが、オブジェクトそれ自身をプロセス間通信でやりとりできないため、オブジェクトとプロセス間通信データの間で変換のためのオーバヘッドが生じる。これについては、オブジェクト指向のオペレーティングシステムの構築等を検討する必要がある。

7．3　応答性

サービスプリミティブ毎にタイムトレースを表示できるようにしたので、おおよそのアプリケーションプログラムの処理速度は把握できる。しかし実際の処理速度は、ターゲット機の性能に依存するため、ターゲット機に実装してから性能評価する必要がある。

7．4　試験支援環境

シミュレータを利用した試験には次の利点がある。

（1）一貫したソフトウェア開発

プログラミングから試験まで一台のワークステーション上で実行できる。シミュレータを利用してほとんどの試験ができるため、移植後のバグに関しては移植上のトラブルに局所化できる。

（2）異常シーケンスの試験

試験において、特に異常シーケンスの試験を行なう場合、セションシミュレータの持つ異常シーケンス発生機能を利用することが可能であり、きめ細かな試験が可能である。

（3）テスタの開発

下位テスタの開発をシミュレータを利用して行える。実際の通信回線を利用した下位テスタとシミュレータ上の下位テスタはまったく同じものが利用可能となる。

8．おわりに

ＯＳＩ高位層プロトコルを実装するためのソフトウェアを効率的に開発するために、セションシミュレータを開発した。

このシミュレータを利用した試験方法について考察を加えた。

今後は、試験支援系としての利用形態での、より使い易いユーザインタフェースの実現を図ってゆきたい。

参考文献

［1］ISO : Information Processing Systems - OSI Basic Reference Model, ISO 7498(1984).

［2］荒木，牛島：通信システムのＡｄａによる統合的開発事例，情報処理，Vol.27, No.3, pp244-253 (1986).

［3］白鳥，高橋，野口：ＮＥＳＤＥＬ：プロトコル向き仕様記述言語とその応用，情報処理学会論文誌，Vol.26,No.6,pp1136-1144(1985).

［4］Bochmann,G.v.:Usage of Protocol Development Tools:The Result of a Survey; Protocol Specification,Testing and Verification VII An International Symposium(1987).

［5］ISO : Information Processing Systems - OSI Basic connection oreiented session service definition(1987).

［6］勝山、佐藤、中川路、水野：オブジェクト指向型言語spiceCによる通信ソフトウェアの開発、情報処理学会研究会、マルチメディアと分散処理、３３－５(1987)。

［7］ISO : Information Processing Systems - OSI Conformance Testing Methodology and Framework,DP9646(1987).

［8］中川路、勝山、水野：ＯＳＩ　ＦＴＡＭの実現、情報処理学会研究会、マルチメディア通信と分散処理、３５－２(1987)。