

LOTOS 実行系を用いた

1 G-5 OSIトランザクション処理用ソフトウェアの実装と評価

野村 眞吾

長谷川 亨

加藤 聰彦

鈴木 健二

国際電信電話株式会社 研究所

1. はじめに

筆者らは、仕様記述言語によるプロトコル仕様に基づく通信システム開発を目的として、LOTOSによるプロトコル仕様を、C言語のプログラムに変換して実行するLOTOS実行系を作成している^[1]。本実行系は、高速処理が可能なプログラムを生成できるという特徴を持つ^[2]。そこで、通信システム開発への適用可能性を評価するために、複数の通信システム間において関連する処理の一貫性を保つ際に使用されるOSIトランザクション処理用(TP)ソフトウェアを、本実行系を用いて実装した。実装したTPソフトウェアを用いた相互接続実験と性能評価実験の結果、本ソフトウェアが実システムとの通信に耐え得ることが明らかとなった。本稿では、LOTOS実行系を用いたTPソフトウェアの実装とその評価について述べる。

2. LOTOS 実行系の概要

本実行系は、トランスレータとスケジューラにより構成される^[1]。トランスレータは、LOTOSで記述されたプロトコル仕様のうちの動作記述部を、スケジューラ機能と呼び出すCプログラムに変換する。スケジューラは、仕様に定義されるプロセスを擬似並列プロセスとして実行し、イベントによるプロセス間の同期処理とプロセスの切替えを効率的に実現する。

LOTOSでは、プロセス間の同期において、やりとりされるデータの型(ソート)と値が一致することを確認する必要がある。このため、トランスレータにより変換されたプログラムでは、仕様で宣言された変数及び値を、ソートの識別子と値を要素として持つ構造体(デスク립タ)により実現している。

本実行系を用いた通信システム開発においては、実装者は、トランスレータを用いて仕様を変換するとともに、以下のマニュアルコーディングを行なう必要がある。

(1) 抽象データ型を用いたソートとその操作(オペレーション)の記述に対して、オペレーションを実現する関数の本体部をコーディングする。特に、LOTOSによるプロトコル仕様では、PDU(プロトコルデータ単位)を抽象データ型で定義するのが一般的であるため、符号化/復号などのPDU処理は、実装者がコーディングすることになる。これらのコーディングにおいては、オペ

レーション関数の引数及び戻り値に対して、前述のデスク립タを使用する必要がある。

(2) 実装者は、変換されたプログラムと外部プログラムとの相互作用をコーディングする。この相互作用は、仕様においては、外部から観測可能なゲートにおけるイベントとして記述される。トランスレータはこれらのゲートに対して、動作の定義されていない擬似並列プロセスを生成するため、実装者はその本体部に、プログラム間の通信機能を用いてイベントに対応する入出力のコーディングを行なう。

3. TPソフトウェアの実装

本実装は、これまでVAX/VMSで動作していたスケジューラをSPARC/UNIXに移植し、SUNワークステーション上で行なった。

3.1 TP仕様の変換

本実装においては、ESPRIT IIプロジェクトの一つであるLotosphereで開発されたLOTOS仕様^[3]を使用した。本仕様は、DIS版OSI TPのカーネル機能単位、コミット機能単位、共通機能単位を規定しており、その規模は全体が3,870行、動作記述部が1,611行である。本仕様をトランスレータにより変換した結果、4,407行のプログラムが得られた^[4]。

3.2 マニュアルコーディング

3.2.1 抽象データ型のオペレーション関数の実現

TP仕様中に定義された203個のデータ型操作に対して、オペレーション関数の本体部を作成した。このうち175個については、PDU処理に関するものであった。

OSIの応用層プロトコルでは、PDUの構造はASN.1により定義される。ASN.1の符号化規則に従ったPDU処理を行なうオペレーション関数の実現は、筆者らが別に作成したASN.1コンパイラ^[5]を使用することとした。このコンパイラの生成する符号化/復号プログラムは、ASN.1記述に対応する構造を持つ内部変数を用いて、符号化前または復号後のPDUの値を保持する。そこで、符号化前または復号後のPDUに対応する変数や値に対応するデスク립タを、PDUを表すソートの識別子と、符号化/復号プログラムの内部変数を指すポインタから構成することとした。これにより、符号化/復号を行なうオペレーション関数を、ASN.1コンパイラが生成した符号化/復号プログラムを呼び出すだけで実現することができた。また、符号化前または復号後のPDUの値を参照/設定するオペレーション関数については、LOTOS実行系の使用データ構造と、符号化/復号プログラムの内部変数のデータ構造の違いを考慮に入れて作成した。

"Implementation and Evaluation of OSI TP Software using LOTOS Execution System"

Shingo NOMURA, Toru HASEGAWA, Toshihiko KATO and Kenji SUZUKI

KDD R & D Laboratories

3.2.2 外部プログラムとの相互動作の実現

(1) ACSE 以下のプロトコル処理との相互動作

本実装においては、ACSE 以下のプロトコル処理を、筆者らが先に開発した OSI7 層ボード^[6]を用いて実現することとした。このため、ACSE およびプレゼンテーション層 (PL) に相当するゲートに対応した擬似並列プロセスの本体部に、OSI7 層ボードの提供する ACSE と PL のサービスプリミティブの作成/解析処理およびデバイスドライバを介したデータの読み書きを実現した。

(2) TP ユーザプログラム

ダイアログの開始 (TP-BEGIN-DIALOGUE) 要求の送信/指示の受信、データ (TP-DATA) の送信/受信などを実行する TP ユーザプログラムを作成した。ユーザに相当するゲートに対応する擬似並列プロセスの動作として、UNIX のソケットを使用したデータの入出力を実現した。

4. 評価実験

実装した TP ソフトウェアを評価するために、相互接続実験とデータ転送の性能評価実験を行なった。

4.1 相互接続実験

実装した TP ソフトウェアが正しく動作することを確認するために、別に開発された TP ボード^[7]と相互接続実験を行なった。TP ボードを装着したパソコンとの間で、ダイアログを開始してデータを転送する正常シーケンスの動作を確認した。

4.2 データ転送の性能評価実験

SPARCserver 670MP と 630MP (CPU: SPARC 40MHz×2) 上に実装した TP プログラム間で、ダイアログの開始に必要な時間およびデータ転送の性能を測定した。両ワークステーション間は、伝送速度が 2Mbps の擬似回線を用いて接続した。通信においては、トランスポート層の最大 TPDU サイズと、ネットワーク層の最大パケットサイズをともに 2,048 バイトとし、トランスポート層でセグメンティングを行なった。

TP-BEGIN-DIALOGUE 要求プリミティブを発行し、対応する TP-BEGIN-DIALOGUE 指示プリミティブが通知されるまでのダイアログの開始時間は、平均 2.3 秒であった。また、1 回の転送データ長を変更して、400K バイトのデータを転送した場合の性能を表 1 に示す。

表 1: データ転送性能の測定結果

データ長 (バイト)	処理時間 (秒)	スループット (K ビット/秒)	処理数 (PDU/秒)
1,000	81.0	39.5	4.93
2,000	40.7	78.6	4.91
4,000	20.7	155	4.83

さらに、これらの通信中に生成/消滅するプロセス数と同期回数を表 2 に示す。

5. 考察

(1) 相互接続実験と性能評価実験を通して、本実行系を用いて実装した TP ソフトウェアが、実システムとの通信に耐え得る性能を実現することを明らかにした。

表 2: 通信におけるプロセス数と同期回数

	ダイアログの開始		データ転送	
	起動側	応答側	送信側	受信側
生成プロセス数	85	77	3	6
消滅プロセス数	9	12	0	0
同期回数	20	13	4	6
平均同期プロセス数	2.65	2.92	2.25	2
最大同期プロセス数	4	4	3	2

(2) 今回は Lotosphere で開発された TP 仕様を利用したが、ISO の標準文書にも LOTOS 仕様が付添されるなど、LOTOS 仕様が容易に入手可能となりつつある。これらの LOTOS 仕様から本実行系を用いてソフトウェアを実装することにより、相互接続性を試験するためのリファレンスシステムの開発を行なうことができる。

(3) 性能評価実験の結果では、データ長に依存せず 1 秒間に平均 5 個のデータを処理している。これは、LOTOS のプロセスと同期を実現する処理が性能を決定しているためと考えられる。本実行系は、効率の高いプロセスと同期の処理を実現しており^[1]、このため実システムとの通信に耐え得るソフトウェアが実装可能であったと考えられる。

(4) PDU 処理のコーディングにおいては、デスクリプタに ASN.1 コンパイラが生成する符号化/復号プログラムの内部変数を指すポインタを持たせることにより、既存の ASN.1 コンパイラを使用し、PDU の符号化/復号に関するオペレーション関数を容易に実現することが可能となった。また、外部プログラムとの相互動作のコーディングにおいては、本実行系が生成する擬似並列プロセスに着目するのみでよく、容易に行なうことができた。

6. まとめ

本稿では、筆者らが開発した LOTOS 実行系を用いて、TP ソフトウェアを実装した結果について述べた。本ソフトウェアを用いた相互接続実験と性能評価実験を行ない、本実行系を用いて、パケット網や ISDN を介したデータ通信が可能な通信ソフトウェアを実装できる見通しを得た。最後に日頃御指導頂く KDD 研究所 浦野 所長、眞次次長、並びに御討論頂いた交換グループ若原 リーダに感謝します。

参考文献

- [1] S. Nomura, T. Hasegawa, T. Takizuka, "A LOTOS Compiler and Process Synchronization Manager", PSTV, X, IFIP, 1990.
- [2] S. Nomura, T. Takizuka, T. Hasegawa, R. Greve, "Performance of LOTOS Execution System", Conf. of IPSJ, 6P-08, Mar. 1993.
- [3] R. Greve, I. Widya, "TP Protocol Version 2.0: a two-way synchronization version", Lotosphere project ESPRIT 2304 Lo/WP3/T3.1/UT/N0016/V3, UT, PTT Research, 1992.
- [4] S. Nomura, T. Takizuka, T. Hasegawa, R. Greve, "Evaluation of LOTOS Execution System with an OSI TP protocol specification", Conf. of IPSJ, 4V-06, Oct. 1992.
- [5] T. Hasegawa, S. Nomura, T. Kato, "Implementation and Evaluation of ASN.1 compiler", Journal of IPSJ, Vol.15, No.2, 1992.
- [6] 井戸上, 加藤, 鈴木, "OSI7 層ボードの実装と評価", マルチメディア通信と分散処理研究会, 61-28, July, 1993.
- [7] 杉山, 小花, 鈴木, "OSI TP (トランザクション処理) ボードの実装と評価", 1993 年春季信学大会.