マルチメディア通信と分散処理　46－3
（１９９０．　７．　１２）

クライアント／サーバ・モデルに基づく LOTOS 仕様記述支援システムの設計

辻　宏郷　　（三菱電機 (株) 情報電子研究所）
佐藤　嘉一　　（沖電気工業 (株) コンピュータシステム開発本部）
内山　光一　　（(株) 東芝 情報通信システム技術研究所）
小野　昌秀　　（沖電気工業 (株) コンピュータシステム開発本部）
五ノ井　敏行　（富士通 (株) 情報システム事業本部）
田中　功一　　（三菱電機 (株) 情報電子研究所）
藤田　朋生　　（日本電気 (株) 基本ソフトウェア開発本部）
山中　顕次郎　（日本電信電話 (株) ＮＴＴソフトウェア研究所）
大蒔　和仁　　（電子技術総合研究所 情報アーキテクチャ部）

OSI プロトコルやサービスの仕様を形式的に記述するためにLOTOS が開発され、これを用いた仕様記述が行われる様になってきた。しかし、従来から存在する支援ツールの多くが、LOTOS のサブセットを対象としており、実際の仕様開発に適用できなかった。我々は、LOTOS 研究会を開催し、記述実験や調査研究を行ってきた。この経験を踏まえて、現実のプロトコルやサービス開発に適用可能な、LOTOS 仕様記述支援システムの設計を行った。このシステムは、分散環境上で動作する、クライアント／サーバ・モデルに基づく支援システムであり、モジュール単位の仕様記述や、非決定性イベントの自動選択実行機能を備えている。

Design of LOTOS Support System based on Client-Server Model

Hirosato TSUJI　　(Mitsubishi Electric Corporation)
Yoshikazu SATO　　(Oki Electronic Industry Co.,Ltd.)
Mitsukazu UCHIYAMA　(Toshiba Corporation)
Masahide ONO　　(Oki Electronic Industry Co.,Ltd.)
Toshiyuki GONOI　　(Fujitsu Limited)
Kouichi TANAKA　　(Mitsubishi Electric Corporation)
Tomoo FUJITA　　(NEC Corporation)
Kenjirou YAMANAKA　(Nippon Telegraph and Telephone Corporation)
Kazuhito OHMAKI　　(Electrotechnical Laboratory)

LOTOS was developed to describe the formal specification of OSI protocols and services. Recently several descriptions of LOTOS specification have been published. But most of usual support tools of LOTOS only accept the subsets of LOTOS , and they couldn't be applied to such real protocol specifications. We , members of LOTOS reserach group , have been writing an experimental desciptions , and designed LOTOS Support System called "LOTOS Server". LOTOS Server is running on the distributed environments , based on Client-Server Model. Moreover our system supports the module specification description of LOTOS , and the automatic selection and execution of the nondeterminate events.

1 はじめに

ISO (国際標準化機構) によるOSI (開放型システム間相互接続) の標準化が進展している。そこで、通信プロトコルやサービスの仕様を、曖昧さを排して記述するためのFDT (形式記述技法) が注目されるようになってきた。ISOではOSIの仕様記述に対する適用を目的として、仕様記述言語LOTOS [1] を開発した。

最近では、LOTOSを用いたOSIの仕様記述の試みが行われており、セション層 [3][4]、トランスポート層 [5][6]、トランザクション処理 [7] 等の形式記述仕様が作成されている。一方で、LOTOSによる仕様記述を支援するために、多くのツールの開発が行われている [8][9][10] 。

我々は、1989年1月から1990年3月の間、**LOTOS研究会**を開催し、LOTOSを用いた仕様記述実験や、調査研究を行ってきた [15] 。
そして、OSI応用層プロトコルの一つであるACSEをLOTOSを用いて記述した [12]。また、OSI 応用層におけるPDUの定義に用いられている抽象構文記法ASN.1 [2] を、LOTOSのデータ型であるACT ONEで利用するための変換方法を検討した [13] 。

今回、これらの検討を踏まえて、**LOTOS仕様記述支援システム**の作成を計画し、このシステムの設計を行った。

以下、第2章で従来の支援系と問題点を述べ、我々の提案する支援系の形態を第3章で示す。また、第4章で具体的な支援系の実現像を示し、さらに、第5章と第6章でこの支援系の持つ特徴を説明する。

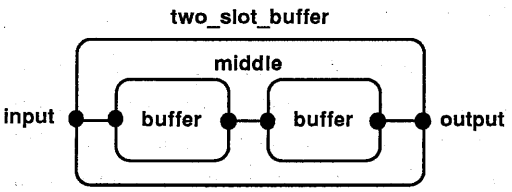

図1 LOTOS仕様のイメージ

2 LOTOSと従来の支援系

2.1 仕様記述言語LOTOS

LOTOS (Language Of Temporal Ordering Specification) は、外部から観測可能なイベントの生起順序を記述することで、対象システムの仕様記述を行う。記述はプロセスと呼ばれるモジュール単位で行い、ゲートと呼ばれる観測可能点で発生するイベントの時間順序を記述する。イベントで発生するデータ部は、抽象データ型を用いて記述を行う。

LOTOSを用いた仕様記述のイメージを図1に、対応する仕様記述例を図2に示す。

2.2 LOTOSの困難性

LOTOSを用いることによって、記述対象となる仕様を、曖昧さを排して定義することができる。その反面、形式的に記述された仕様は、直感的に理解しにくく、仕様を読解する実装者のみならず、記述した設計者本人にとっても、誤解を生じる可能性があった。このため、仕様の理解を援助する支援ツールとして、LOTOS仕様を入力とし、記述された振るまいをユーザと対話的に実行するシミュレータが存在する。既存のツールでは、図式表現を用いたり操作を簡単化することによって、ユーザインタフェースを向上させる試みが行われている。

```
specification two_slot_buffer [input,output] : noexit

type nat is

      sorts  nat
      opns   0     :              -> nat
             s     : nat          -> nat
             _+_ : nat, nat -> nat
      eqns   ofsort nat  forall n, m : nat
             n + 0      = n     ;
             n + s(m) = s(n+m) ;
endtype

behaviour
      hide middle in
            buffer[input,middle]
            |[middle]|
            buffer[middle,output]

where
      process buffer [input, output] : noexit
            :=  input ? x: nat ;
                output ! x ;
                buffer [input,output]
      endproc

endspec
```

図2 LOTOS仕様記述例

また、LOTOSは記述の自由度が高い言語である反面、非決定性イベントや無限分岐、無限ループ等を容易に書くことが可能である。(図3 参照) さらに、データ定義に用いている抽象データ型において、等式の評価を行うための項の再構成は、計算不能であったり、あるいは実行する計算機の性能に大きく依存する。従来のツールの多くは、取り扱い対象を LOTOSのサブセットとして制限を与えたり、人間と対話的に実行して人間の助力を得ることで、これらの問題を回避してきた。

2.3 従来の支援系の問題点

前節で述べた様に、LOTOS は計算機にとって扱いにくい言語である。従来から存在する支援系の多くには、以下の様な問題点が存在した。

- 研究用のプロトタイプであり、処理対象がLOTOS のサブセットに制限されている。
- 実際の通信プロトコル、サービス記述の様な大規模な仕様を扱うことができない。
- 様々なユーザのレベルや好みに応じたインタフェースを提供することができない。
- 対話型のシミュレーションが基本であり、バッチ処理ができない。
- 実行可能な計算機に制限があり、十分な処理能力のある計算機上で使用できない
- 拡張性に乏しく、他の支援ツール開発に際して応用することができない。

従って、従来の支援ツールはLOTOSという仕様記述言語の記述能力を評価し、実際の仕様開発に用いるためには、十分とは言えなかった。

```
process Comp [ port,id ] (n:nat) : noexit :=
    ( choice x:int  []
      port!n ; id!x ; Comp [ port,id ] (n) )
    |||
    Comp [ port,id ] (n+1)
endproc
```

図3 無限分岐するプロセス記述例

3 クライアント／サーバ・モデルに基づく支援システム

3.1 支援システムの基本方針

我々は、LOTOS による仕様記述を支援するシステムとして、LOTOS 仕様のシミュレータを中心とした支援系を作成することとした。システムの設計にあたっては、実際に従来のツールを用いた仕様記述の経験を踏まえて、以下の様な基本方針に基づいて設計を行った。

第一に、現実のプロトコルやサービス開発に適用可能な支援系とすることである。このためには、LOTOSの国際規格で許されている全ての構文や意味を取り扱うことができる必要がある。また、現実的な仕様を、一定の時間内で処理できる必要がある。

第二に、ユーザのレベルに合わせたインタフェースを提供することである。初心者には容易なインタフェースを、熟練者には作業効率を重視した動作環境を提供したい。例えば、小規模な仕様は対話的に実行し、大規模な仕様は後述する非決定性イベントの自動選択実行機能を用いてバッチ処理が実行できるようにする。

第三に、分散環境上で動作することである。本システムは、利用者が保有する計算機資源を効率的に利用して実行できる構造を持つ必要がある。と同時に、将来のLOTOS仕様記述支援ツール開発における利用を考慮して設計を行う。

3.2 LOTOS サーバとクライアント

我々は、前節で述べた要求を満たすLOTOS仕様記述支援システムを作成するにあたり、クライアント／サーバ・モデルに基づくプログラムでこれを実現することとした。すなわち、サーバ側として、LOTOS 仕様の構文解析・静的意味解析・推論規則の適用など、基本的な処理機能を提供する **LOTOS サーバ**を作成する。その一方で、**クライアント側アプリケーション**として、仕様の解析を行う**アナライザ**や、仕様のシミュレーションを行う**シミュレータ**を作成する。

この支援システムの典型的な実行例を図4に示す。もちろん、1台の計算機上でサーバ、クライアント両方のプログラムを実行すれば、従来のツール同様、単一計算機上の環境で使用することが可能である。

また、サーバが提供する基本的な処理機能を利用して、新たなLOTOSの支援ツールを作成することが容易となる。サーバと周辺アプリケーションから構成される支援システムを図5に示す。次章以降では、支援システムの中枢をなすLOTOSサーバを中心に説明する。

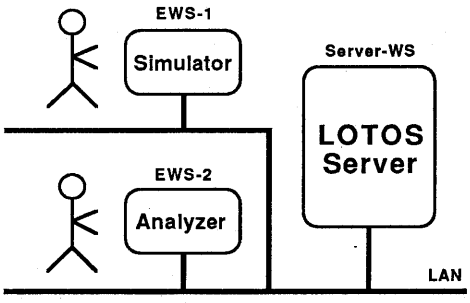

図4　LOTOSサーバ使用形態

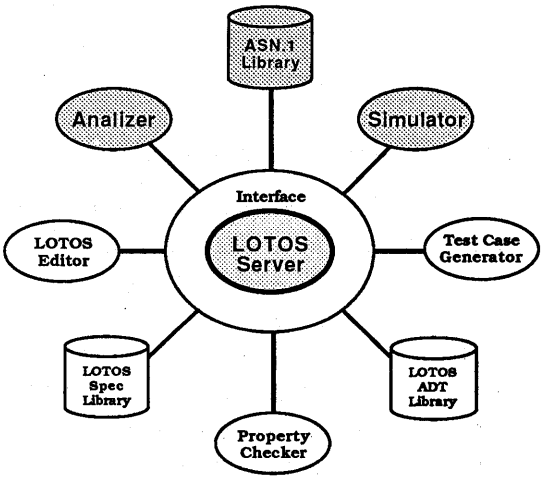

図5　サーバと周辺アプリケーション

3.3　実現上の利点

LOTOS支援システムをクライアント／サーバ方式で実現することによって、以下の様な利点が得られる。

1) LOTOS仕様の持つ困難性を、サーバとクライアントが分担して解決し、計算機の負荷を分散することができる。例えば、サーバを処理能力の高い計算機で実行し、クライアントをユーザインタフェースの良い計算機で実行する。

2) LOTOS支援系を実現する上での基本的な解析機能と、ユーザインタフェースを分離する。従って、ユーザのレベルや好みに合わせたインタフェースの選択が可能である。

3) 支援系に必要な基本機能とインタフェースを、個別に設計したり、独立に改良することが可能となる。特にクライアントを改良することで、支援機能の拡張やインタフェースの向上が容易に実現できる。

4) LOTOS支援ツールを実現する際に必要な機能が、サービスとして提供される。このサーバ機能を組み合わせて、新たな支援ツールであるクライアントを作成することが可能である。

4　LOTOSサーバ

4.1　LOTOSサーバの要求仕様

LOTOSサーバの基本設計思想は、ISO/IS 8807に従って記述された仕様を忠実に実行する抽象機械を実現することである。すなわち、LOTOS仕様（テキスト）に対して、国際規格で定義された構文規則、意味規則、推論規則等を忠実に適用するシステムである。

また、サーバは処理能力の高い計算機上で実行することを前提としており、複数のクライアントに対応できる必要がある。

4.2　サーバが提供する基本的機能

LOTOSサーバはユーザ（クライアント）に対して、以下の基本的な機能を提供する。

○ LOTOS仕様（テキスト）の登録・削除

○ 構文解析、静的意味解析、簡単化
○ LOTOS 中間言語への変換
○ 中間言語のロード／セーブ
○ シミュレーション（動的意味解析）

ユーザは、クライアントを介してのみ、サーバを利用することができる。クライアントとして構造エディタを考えた際の、LOTOSサーバの利用イメージを図6に示す。また、ユーザのサービス要求に対するサーバの内部動作を図7に示す。

4.3 サーバの特徴

LOTOS サーバは、従来のシミュレータと比較して、以下に示す特徴を持っている。

◎ マルチクライアント・サーバ
◎ モジュール単位仕様記述のサポート
◎ 非決定性イベントの自動選択実行機能
◎ 汎用的な中間言語の利用

我々は、モジュール単位の仕様記述とイベントの自動実行を行うために、LOTOS の意味を国際規格の上位互換を満たす様に拡張した。これらは次章以降で説明する。また、サーバで用いている中間言語については別途報告する [14]。

基本的にサーバは、LOTOSの国際規格で定義されている意味を、全てサポートする方針である。しかし、抽象データ型の等式の代数的な解釈、すなわち始代数に基づく同値類による意味付けは、計算機上実現が困難であるため、等式を左辺から右辺への書き換え規則とみなして、項書き換え系による処理を行う。但し、複数の書き換え戦略を扱うことを検討している。

4.4 サーバの利用例

LOTOS サーバを利用するクライアントとして、以下の様なプログラム例を挙げることができる。

○ アナライザ＆デバッカ
○ シミュレータ
○ 構造エディタ
○ 試験系列ジェネレータ
○ プロパティチェッカー

また、複数のサーバを駆使して、通信する複数仕様のシミュレーションを行うことができる。現在は、クライアントプログラムとして、

仕様記述→解析→シミュレーション

を一体化し、モジュール単位の仕様記述を可能とする、構造エディタの開発を計画している。

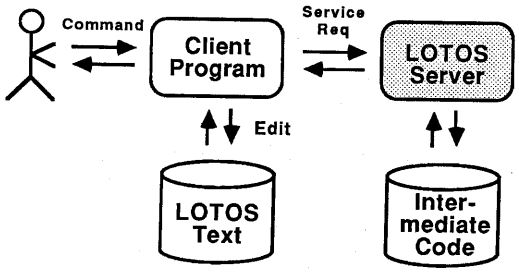

図6 ＬＯＴＯＳサーバの利用イメージ

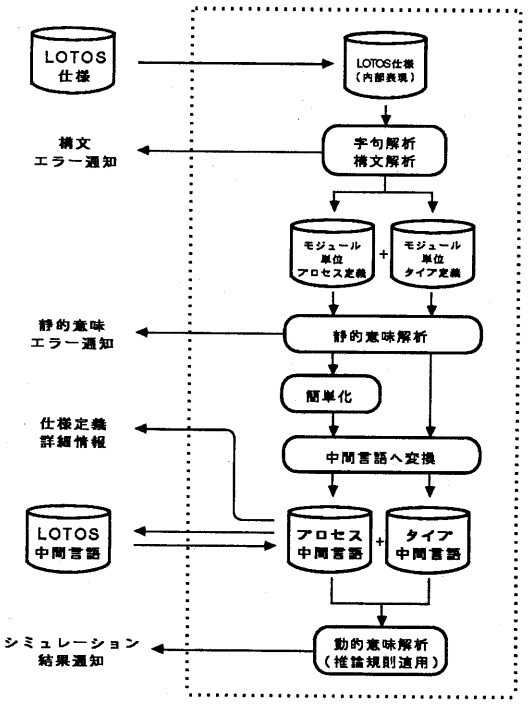

図7 サーバの内部動作

5 モジュール単位の仕様記述支援機能

5.1 モジュール単位の仕様

LOTOS では、プロセス、データ型といった単位で仕様記述を行い、これらのモジュールを組み合わせて全体の仕様とする。LOTOS サーバは、このモジュール単位の仕様記述を支援するために、サブプロセス、サブデータ型単位の解析・シミュレーション機能を提供する。

5.2 仕様の構造と記述単位

LOTOS 仕様の大まかな構造に注目すると、仕様全体とプロセス定義、データ型定義は、それぞれ 以下の様になる。

```
spec  ::=  spec-id     仕様名と引数
           [{type}]    グローバルデータ型
           init-BE     初期動作式
           [{proc}]    サブプロセス
           [{type}]    サブデータ型

proc  ::=  proc-id     プロセス名と引数
           BE          動作式
           [{proc}]    サブプロセス
           [{type}]    サブデータ型

type  ::=  type-id     タイプ名
           [{sort}]    ソート
           [{opns}]    演算子
           [{eqns}]    等式
```

従って、**仕様全体** (spec)、**プロセス** (proc)、**データ型** (type) 単位の仕様の登録・追加・削除機能を提供することで、モジュール単位の LOTOS仕様記述が可能となる。また、**仕様名** (spec-id) のみ、**初期動作式** (init-BE) のみの登録・削除機能を提供することで、モジュールの組合せを試行錯誤しながら仕様記述することが可能となる。

5.3 モジュール単位記述の効果

本機能を利用することで、記述途上の不完全な仕様に対しても、有効な領域をユーザが明示することで、解析・シミュレーションを行うことが可能である。また、モジュール単位で仕様を記述しておき、これらの仕様を組み合わせて適当な初期動作式を与えて、シミュレーションを行うことができる。従って、他のプロセスに制約を与える並列オペレータやプロセスを、試行錯誤で記述することが容易となる。すなわち、LOTOS の記述スタイルの特長である制約指向 [11] による仕様記述を支援することができる。

以上の様に、モジュール単位の仕様サポート機能は、LOTOS に対して**構造化仕様記述**のスタイルを提供する。モジュール単位で記述した仕様をライブラリ化し、部品化した仕様を再利用することが容易となる。この部品仕様をサポートするクライアントを作成することによって、C 言語における **include** 文 や分割コンパイルの様な仕様記述スタイルを可能とする。

6 非決定性イベントの自動選択実行機能

6.1 シミュレーションの基本機能

LOTOS では、記述対象の仕様において、発火するイベントの時間順序を記述する。従って、LOTOS仕様は、初期動作式をルートとするイベント木とみなすことができる。図 8 は、図 2 の仕様から得られる状態遷移木の一部である。

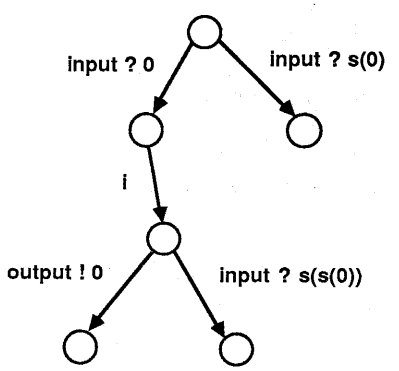

図 8　仕様から得られる状態遷移木

LOTOS仕様のシミュレーションとは、この状態遷移木を作成し、その木の上を走査することである。ここに、状態とは動作式と変数値の組合せであり、遷移とはイベント、すなわちゲートと抽象データ値の組合せを意味する。

以下は、木構造のノード（状態）において、サーバがシミュレーションのために提供する基本的な機能である。

1）現在の状態（動作式と発火可能なイベントのリスト）の表示
2）イベントの選択実行
3）記録した状態への移動
4）状態遷移の履歴を消去
5）状態遷移の履歴を表示
6）初期状態から現在の状態までのパス表示
7）ヘルプ機能
8）シミュレーションの終了

6.2 動的解析情報の提示法

サーバは、シミュレーション実行時に、動的意味解析情報を、クライアントの要求に応じて提示する。この節では、クライアント側から選択可能な設定条件を紹介する。

1）動作式とリラベリング情報の提示法

a) リラベリング後の動作式として提示

x ; exit || x ; exit

b) 動作式＋リラベリング情報として提示

a ; exit || b ; exit [x/a , x/b]

上記の例では、動作式とリラベリング情報の組合せとして提示することで、ゲート x においてイベントが同期できない理由が明らかになる。

2）発火イベントの提示法

a) 全てのイベント＋同期条件等を詳細通知
b) 全てのイベントを通知
c) **i** イベント以外の全てのイベントを通知
d) 指定ゲートにおけるイベントのみを通知

この様に、必要なゲートのイベントの情報のみを取り出すことができる。

6.3 非決定性イベントの選択と実行

LOTOS仕様では、イベントの発火可能性のみを記述するので、ある状態において、発火可能なイベントが複数存在し、仕様からはその選択が決定できないことがある。これに対して、LOTOSサーバは、イベントをシミュレータが自動選択し、非対話的に実行する方法を提供する。

1）イベントの選択実行方法

a) 1イベントごとに停止して、ユーザのの選択を要求 （ステップ実行モード）
b) 発火可能なイベントが1種類の場合は自動選択し、複数のイベントが発火可能な点で停止してユーザの選択を要求 （半自動実行モード）
c) ユーザが指定した規則によって自動選択し、ブレークポイントで停止 （自動実行モード）

2）ブレークポイントの指定法

自動選択実行時には、ユーザが指定したポイントまでは停止しない。ブレークポイントの指定には、以下のような方法がある。

a) 仕様中のイベントで指定
b) イベント（ゲート名＋値）で指定
c) 以後発火するイベント数で指定

6.4 自動実行のための LOTOS の拡張

LOTOS において、イベントの発火を決定するのは環境 である。しかし、発火可能なイベントが複数存在する場合、選択を行う指針は仕様中に記述されていない。この非決定性イベントを自動選択するために、仕様中に、特定の文字列をもったコメントでマーキングを行い、自動実行の指針を記述できるようにする。すなわち、イベント発生の重み付け値を記述することで、複数イベントから一定の規則による選択を可能とする。なお、この記述はコメントで表現するので、LOTOSの構文そのものを変更していない。以下に、重み付け記述の例を示す。

```
P1 (* 0.4 *)   []    (* 0.6 *) P2
P1 (* 0.7 *)   |||   (* 0.3 *) P2
P1 (* 99.5 *)  [>    (* 0.05 *) P2
```

この**重み付け値**を元に、イベント発火可能性値を計算し、以下の２つの方法で選択実行する。

1) **計算値の比率に従って、乱数でイベントを選択実行する。** 実行する度に、不規則な、すなわち再現性のない自動選択を行う。
2) **計算値の順位に従って、イベントを優先実行する。** 順位の指定方法によって同一の、すなわち再現性を持った自動選択を行う。

また自動実行モードにおいては、重み付けの記述が存在しない場合、環境と同期する全てのイベントが等確率に起こり得るものと見なして、選択実行を行う。

7 おわりに

LOTOS仕様記述を開発する上で、基本的な解析・シミュレーション機能を提供する、支援システムの設計を行った。このシステムは、様々なユーザに対応し、今後の拡張性を考慮するために、分散環境上で動作するツールとした。また、LOTOS仕様の困難性を解決するために、幾つかの意味の拡張を試みた。今後は本システムを作成し、これを用いてLOTOSの仕様記述能力の評価を行う予定である。

謝辞

Twente 大学の Prof. Ed Brinksma 、ならびに東北大学の 高橋 薫 氏には、貴重な御意見を頂いた。ここに記して感謝致します。また、LOTOS研究会は、多くの方々の協力により開催することができた。参加して下さった各社関係者の方々、会議室使用の便宜を図って頂いた (財) 情報処理相互運用技術協会に感謝致します。

参考文献

[1] ISO/IEC : OSI - LOTOS - A Formal Description Technique based on the Temporal Ordering of Observational Behaviour, ISO 8807 (1989).
[2] ISO/IEC : OSI - Specification of Abstruct Syntax Notation One (ASN.1), ISO 8824 (1987).
[3] ISO/IEC : OSI - Formal Description in LOTOS of OSI Connection-Oriented Session Service, ISO TR9571 (1989).
[4] ISO/IEC : OSI - Formal Description in LOTOS of OSI Connection-Oriented Session Protocol, ISO TR9572 (1989).
[5] ISO/IEC : OSI - Formal Description in LOTOS of OSI Connection-Oriented Transport Service, ISO DTR10023 (1989).
[6] ISO/IEC : OSI - Formal Description in LOTOS of OSI Connection-Oriented Transport Protocol, ISO DTR10024 (1989).
[7] ISO/IEC : OSI - Distributed Transaction Processing - Part 3 : Protocol Specification, ISO DIS10026-3 (1989).
[8] J.Tretmans : HIPPO - LOTOS simulator, The Formal Description Technique LOTOS, (1989).
[9] R.Guillemot, M.Haj-Hussein and L.Logrippo : Executing Large LOTOS Specifications, Protocol Specification, Testing and Verification VIII, pp.399-410 (1988).
[10] J.P.Briand, M.C.Fehri, L.Logrippo, A.Obaid : Executing LOTOS Specifications, Protocol Specification, Testing and Verification VI, pp.73-84 (1987).
[11] E.Brinksma : Constraint-oriented specification in a constructive formal description technique, Stepwise Refinement of Distributed Systems Proc. REX Workshop, (1989).
[12] 内山、藤田、他 : “ACSE の LOTOS による記述の試み”, 情報処理学会第 46 回マルチメディア通信と分散処理研究会, (1990) 発表予定 .
[13] 五ノ井、他 : “ASN.1 から LOTOS ADT への変換方法”, 情報処理学会第 46 回マルチメディア通信と分散処理研究会, (1990) 発表予定 .
[14] 佐藤、他 : “LOTOS の汎用的な中間言語”, 情報処理学会第 46 回マルチメディア通信と分散処理研究会, (1990) 発表予定 .
[15] 大蒔、他 : “哲学者の食事問題の LOTOS による記述実験”, 情報処理学会ソフトウェア工学研究会 66-4, (1989).
[16] 佐藤、川口、高橋、白鳥、野口 : “ SAL : LOTOS 仕様の意味解析支援システム”, 情報処理学会第 40 回全国大会 4N-4,5 (1990).
[17] 野村、長谷川、瀧塚 : “LOTOS 実行系の並列処理環境”, 情報処理学会ソフトウェア基礎論研究会 29-4, (1989).
[18] 辻、高橋、白鳥、野口 : “LOTOS 解釈機構の設計と実現”, 電子情報通信学会研究会 IN89-58, (1989).
[19] 高橋、白鳥、野口 : “LOTOS 解釈機構の構成”, 電子情報通信学会研究会 IN87-107, (1987).